COSC460

Research Project

Department of Computer Science

University of Canterbury

EFFICIENT CODING OF

THE DANTZIG-WOLFE DECOMPOSITION (LINEAR PROGRAMMING) ALGORITHM

C.H. OO

October, 1978.

INDEX

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr C.A. de Kluyver for his help, encouragement and advice with all aspects of this research project.

CHAPTER I

INTRODUCTION

I INTRODUCTION

The Dantzig-Wolfe decomposition (linear programming) principle published in 1960 involves the solving of large-scale mathematical programming problems of particular structure. Large practical problems of this type typically involve many constraints and a large number of variables. For instance, a manufacturer who manufactures various types of household items (thus having many decision variables) may be faced with a multiplant production and distribution problem, needing to maximize profits subject to many constraints such as factory capacities, market potential, raw material availability, budgetary limitations and legal requirements; of which the coupling constraints (constraints used to link the common resources) may arise from common budgetary limitations on the plants, from common capital used for expansion, or from demands for products whose production involves more than one plant.

Often a linear program will be large only because it deals with subsystems with similar types of structure, repeated either in time or location, with the subsystems having relatively few items in common. The Dantzig-Wolfe decomposition principle has been found to be most efficient when applied to problems whose coefficient matrices have angular structure (i.e. one or more independent blocks linked by coupling equations). However, solving this type of general large-scale problem using manual calculations (even with the help of programmable calculators) is totally out of the question. This leads to the aim of this research which follows.

The objective of this project is to implement an efficient ALGOL coded version of the Dantzig-Wolfe decomposition algorithm (derived from the principle basically) so as to enable:

1) Large scale linear programming problems in real life to be solved based on the Dantzig-Wolfe decomposition principle with the use of computer.

2) Both undergraduate and postgraduate students to solve related problems by computer in order to have a better understanding of the principle and at the same time provide a tool to assist them in further research in related fields.

CHAPTER II

THEORETICAL ASPECTS

II THEORETICAL ASPECTS

Consider a company having two divisions, producing two and three products, respectively. Each has its own internal resource for production, e.g. labour and machines. The divisions are coupled by the fact that there is a shared resource which both use, for example, a raw material of limited availability. Let division 1 have limited amounts b_1 and b_2 of its internal resources, and division 2 have similar availability b_3. If the limitation on the common resource is b_0, then the problem is formulated as follows: (with aim of minimizing costs)

Objective Function:

Minimize $Z = c_{11}x_{11} + c_{12}x_{12} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23}$

Subject to Constraints:

$$
\begin{aligned}
a_{11}x_{11} + a_{12}x_{12} + a_{21}x_{21} + a_{22}x_{22} + a_{23}x_{23} &\leqslant b_0 \\
b_{11}x_{11} + b_{12}x_{12} &\leqslant b_1 \\
b_{21}x_{11} + b_{22}x_{12} &\leqslant b_2 \\
b_{31}x_{21} + b_{32}x_{22} + b_{33}x_{23} &\leqslant b_3 \\
x_{1j} &\geqslant 0 \quad j=1,2 \\
x_{2j} &\geqslant 0 \quad j=1,2,3
\end{aligned}
$$

where x_{ij} is the quantity of product j produced by division i.

and c_{ij} is the cost of producing a unit of product x_{ij}.

a_{ij} is the quantity of common resources (such as raw material) required to produce 1 unit of product j from division i.

b_{ij} is the quantity of internal resource i (such as machine or labour) required to produce 1 unit of product j.

Z is the value of objective function.

The above constraints consist of two independent subproblems of inequalities, each pertaining to internal operation of one division, and one coupling constraint, introduced by the common resource. This situation could generalize to large problem of following structure:

Minimize (or Maximize) $Z = \sum_{i=1}^{P} C_i X_i$

Subject to

$$
\begin{array}{llllll}
A_1X_1 + & A_2X_2 + & A_3X_3 + & \cdots\cdots\cdots\cdots & + A_pX_p & = b_0 \\
B_1X_1 & & & & & = b_1 \\
 & B_2X_2 & & & & = b_2 \\
 & & B_3X_3 & & & = b_3 \\
 & & & \cdots\cdots & & \cdots \\
 & & & \cdots\cdots & & \cdots \\
 & & & & B_pX_p & = b_p \\
 & & & & X_i \geqslant 0 & i = 1,2,\ldots p
\end{array}
$$

where X_i and b_i are now vectors of variables.

C_i are cost vectors.

and A_i's, B_i's are now matrices.

The above angular type of structure may be solved using the Dantzig-Wolfe decomposition algorithm as follows:

1) Form an equivalent "master program" which essentially consists of the coupling constraints (i.e. $\sum_{i=1}^{P} A_iX_i = b_0$) and the convexity constraints (i.e. $\sum_j W_j = 1,\ W_j \geqslant 0$). The W_j's in the convexity constraints denote the weights of the various proposals (namely, the optimal solutions of various subproblems) to the master program.

2) Using the simplex multipliers or prices as parameters together with the original objective function to form the objective functions for various subproblem sets. (i.e. $B_iX_i = b_i$, i=1,2,...p).

3) Apply Simplex or the Revised Simplex method to obtain optimal solutions for each subproblem set.

4) Entering columns to the master program are generated from the optimal solutions obtained for various subproblem sets.

5) Consider only these entering columns, which could improve the current master solution, as the candidate proposals to the master program and pivot them in the order of merit.

6) The master program is then tested for optimality and new multipliers or prices are computed.

7) Repeat step 2 to step 6 until optimal solution is obtained for the master program.

CHAPTER III

PRELIMINARY INVESTIGATION

III PRELIMINARY INVESTIGATION

At present, there are but a few inefficient programming codes for Dantzig-Wolfe decomposition algorithm. This could be due to the fact that most of the algorithms being described so far are not suitable for coding purposes without modification. Thus, prior to the actual coding of the program, there were a number of areas in the algorithm to be investigated:

1) Finding of solution approaches available in solving both the master program and the subproblem sets, and the selection of best route of approach in terms of minimizing computer core storage and running time. So far, Simplex and the Revised Simplex Method# are commonly used in solving subproblem sets and the master program respectively. A comparison has been made between the two methods as regards to the use of computer (see Appendix E) and Revised Simplex is found to be superior, hence this method is adopted in solving both the various subproblem sets and the master program.

\# (Simplex Method is merely a procedure which converts a system of equations to a canonical form with the introduction of artificial variables thus creating a corresponding basic solution. The optimal solution of the problem is then undertaken in two phases. In phase I, the sum of artificial variables is minimized and if the sum is reduced to zero, the original problem has a feasible solution and phase II is entered. In phase II, the true objective function is optimized. However, if the sum of artificial variables cannot be reduced to zero, then the original problem has no feasible solution. During the process of searching for an optimal solution, at each iteration, it picks a variable to enter the basis and another variable to leave the basis. In the case where no variable is eligible to leave the basis and thus no variable can in turn enter the basis, the original problem is said to have an unbounded solution. Revised Simplex Method is similar to Simplex Method basically except that it only generates the necessary information required.)

2) Selection strategies as regards to whether all possible p proposals (or candidates) should be attached to the master program in a single cycle or only the best proposal (or candidate) which prices out most negatively. The former strategy is found to be reaching the optimal solution within fewer cycles, as compared to the latter, however, the following additional tests are necessary with the adoption of the former strategy:

   a) Test the optimality of master program each time, after the best proposal is pivoted into the master basis though there are still p-1 or less proposals to be considered.

   b) After the best proposal (out of the p or less proposals) has been pivoted into the master basis, the previously known to be second-best proposal may not be the best proposal in this cycle, namely, the order of merit for the remaining proposals are to be computed and best proposal is then to be reselected.

3) The adoption of postoptimality technique in searching for optimal solution for subproblem sets. We have found that after the optimal solutions for subproblem sets have been obtained for the 1st master cycle, the solutions could be retained and served as the starting feasible solutions for the respective subproblem sets in the later master cycles, instead of resolving the subproblem sets each time from scratch (i.e. forming of initial basis, driving out of corresponding artificial variables ..... etc.). The post-optimality performed on the various subproblem sets could speed up the process tremendously. Thus this technique has been included in the algorithm for coding purposes.

There are programs available for solving linear programming Simplex Method and Revised Simplex Method but most of these are written in FORTRAN # and the programme structures are rather rigid due to lack of flexible features available in ALGOL such as declaration of variable bound arrays etc. Since these FORTRAN routines could not be used without appropriate modification, it is considered to be more worthwhile to recode these routines rather than applying the existing modules, in order to maintain the consistency throughout the design of the data structure for the entire program.

(Though there is a Revised Simplex routine available in TEMPO - a mathematical programming package, which is written in ALGOL-liked language. The program module set up is very much package dependent and is not suitable for use without modification).

CHAPTER IV

PROGRAM DESIGN AND DOCUMENTATION

IV PROGRAM DESIGN AND DOCUMENTATION

1) General

The program for the Dantzig-Wolfe decomposition linear programming algorithm is coded in ALGOL language. It is designed to be package-liked and very much user orientated as far as efficiency is concerned (i.e. 'Efficient' in terms of user point of view as regards to how feasible are the various facilities provided by the code in meeting the user requirements).

2) Program Structure

The program essentially consists of 3 main sections:

a) Editing of Input Data

This section handles the editing of input data specified by the user. Information related to the characteristics of the user's problem such as the number of objective functions, the number of coupling constraints and the number of constraints in various subproblem sets etc., are treated as part of the control data, and these specification are later used as a guide in validating subsequent input data. If the amount of input data being read does not meet the respective specification, a relevant error message is printed and the program is subsequently aborted. A list of possible error messages is illustrated in Appendix D.

b) Generating of Solutions for Subproblem Sets

This section has been implemented to obtain the optimal solution for each subproblem set using the Revised Simplex Method. Objective functions for various independent subproblem sets are set up to contain variable parameters so that they are influenced by the simplex multipliers or prices passed down from the master program. Optimal solutions generated for various subproblem sets are saved after each master cycle, and used as the initial feasible solution for the corresponding subproblem set in the subsequent cycle in order to reduce the computer process time.

In case where no optimal solution is obtainable for a particular subproblem set, the program will terminate abnormally with one of the following informative messages printed:

i) "NO COLUMN IN THE TABLEAU IS ELIGIBLE TO LEAVE THE BASIS AFTER N ITERATIONS THUS SUBSYSTEM P HAS AN UNBOUNDED SOLUTION"

OR

ii) "ARTIFICIAL VARIABLE BEING PIVOT INTO THE BASIS THUS SUBSYSTEM P HAS AN INFEASIBLE SOLUTION"

The following modules are set up to handle various functions in this section:

a) ADDSLASURART

This procedure is used to convert a system of constraint equations in each subproblem set to its equivalent canonical form by introducing the slack, surplus and/or artificial variables depending on the status of each constraint.

b) SETUPBASIS

This procedure is used to set up the initial basis inverse for various subproblem sets before they are solved for the first time.

c) ZJMINUSCJ

This procedure computes the $Z_j - C_j$ values for all vectors not in the basis and thus detect for optimality for various subproblem sets. If optimality has not been reached it determines the entering vector and leaving vector appropriately.

d) UPDATE

This procedure updates the basis inverse of various subproblem sets once the pivot element is found.

e) MATRIXREINVERSION

This procedure is used to reinvert the basis inverse of corresponding subproblem set in order to prevent the loss of accuracy in data computed due to accumulation of rounding off errors.

c) <u>Optimizing of the Master Program</u>

This section has been implemented to search for an optimal solution (if any) for the entire user problem. It is set up to be closely linked with the last section so that prices (or simplex multipliers) could be passed down via the respective objective functions. Revised Simplex is again used in optimizing the master program except that the entering columns are now formed by using the proposals sent up by the various subproblem sets. All eligible proposals are pivoted into the master basis in order of merit and best candidate is reselected each time.

Similar to last section, one of the following relevant messages will be printed wherever there is an infeasible or unbounded solution to the master program, i.e.

i) "NO INCOMING CANDIDATE COLUMN GENERATED FROM SUBSYSTEMS WHEN IN PHASE I THUS MASTER PROBLEM HAS AN INFEASIBLE SOLUTION"

OR

ii) "NO COLUMN IN THE TABLEAU IS ELIGIBLE TO LEAVE THE MASTER BASIS IN ITERATION T DURING THE PIVOTING OF ENTERING COLUMN FROM SUBSYSTEM P THUS THE PROBLEM HAS AN UNBOUNDED SOLUTION"

There are 3 main modules used in this section, namely,

a) FORMENTERCOL

This procedure is used to form the various proposals based on the solution sets obtained from the respective subproblem sets.

b) PIVOTENTERCOL

This procedure handles the pivoting of eligible candidate proposals into the master and updates the basis inverse accordingly.

c) WRITETODISK

This procedure is used to save the intermediate results onto disk for future processing.

Listing of the source program and flow chart for the code are shown in Appendices B and A respecitvely.

3) Data Structures

A. Arrays

i) Boolean Arrays

a) MP[0:7]

This array stores the flags set for printing various information from the master program in each cycle. (See paragraph Output under sub-heading Input And Output Parameter for details). By default, all these flags are set to TRUE.

b) S [1:NUMBLOCKS, 0:7]

This 2-dimensional array stores the flags set for printing various information on each subproblem set in each iteration (see paragraph Output under sub-heading Input And Output Parameter for details). By default all these flags are set to TRUE.

The 1st dimension of the array indicates the particular subproblem set.

The 2nd dimension of the array indicates the 8 options flags available for the corresponding subproblem set.

ii) Integer Arrays

a) BINDEX [0:BSIZE]

This array stores the indices of variables in the basis for a particular subproblem set. Indices 0 and -1 are used to indicate the permanent basic variables, namely — the values of objective function for phase II and phase I respectively.

b) DUMMYOBJ [0:2 * SIZE + 1]

This array stores the dummy objective values (i.e. -1) for all artificial variables attached to the basis which is to be reinverted using module MATRIXREINVERSION.

c) NUMCOLS [0:NUMBLOCKS]

This array denotes the number of variables in each subproblem set.

d) NUMROWS [0:NUMBLOCKS]

This array denotes the number of constraints in each subproblem set.

f) STATUS [0:NUMBLOCKS, 1:MAXROWNO]

This 2-dimensional array stores the status of each constraint in various subproblem sets.

VALUE	TYPE OF CONSTRAINT
+1 for	$\leqslant$
0 for	=
-1 for	$\geqslant$

The 1st dimension of the array indicates the particular subproblem set.

The 2nd dimension denotes the respective constraint in the corresponding subproblem set.

g) VARSTATUS [1:MAXCOLNO]

This array stores integer values to indicate the status of variables in a particular subproblem set.

VALUE	TYPE OF VARIABLES
1	Real, Non-basic
2	Real, Basic
3	Slack, Non-basic
4	Slack, Basic
5	Surplus, Non-basic
6	Surplus, Basic
7	Artificial, Non-basic
8	Artificial, Basic

h) WEIGHT [2:MSIZE]

This array stores the weights of all proposals in the master basis.

i) XSIZE [1:NUMBLOCKS]

This array stores the total number of variables (excluding surplus and artificial variables) of each subproblem set.

iii) Real Arrays

a) A [1:NUMBLOCKS, 1:MAXROWNO, 1:MAXCOLNO]

This 3-dimensional array stores the coefficients of variables in the constraints of each subproblem set. The 1st dimension of the array indicates the particular subproblem set that is currently referred to.
The 2nd dimension indicates the constraint in that subproblem set.
The 3rd dimension indicates the variable associated with the coefficient, e.g. A[1,2,3] refers to the coefficient associated with the 3rd variable of 2nd constraint in the 1st subproblem set.

b) B [0:BSIZE, -1:BSIZE]

This array is used to store the basis inverse of each subproblem set. The 1st dimension indicates the row of basis inverse.
The 2nd dimension indicates the column of basis inverse.

c) BIGX [2:MSIZE, 1:MVAR]

This array stores the optimal solutions of subproblem sets which form the various proposals whose weights are currently in the master basis.
The 1st dimension of the array indicates the various rows in the master basis (excluding the 0th and 1st rows where permanent basic variables are located).
The 2nd dimension of the array denotes the respective variables in the solution of particular subproblem set.

d) C [1:NUMBLOCKS, 1:MAXROWNO, 1:MAXCOLNO]

This array stores the coefficient matrix of the coupling constraints.
The 1st dimension of the array indicates the block of

coupling constraints which the subproblem set is associated with.

The 2nd dimension denotes the respective coupling constraints.

The 3rd dimension denotes the respective variable in the coupling constraint, associated with the particular subproblem set.

e) CONSTRNAME [0:NUMBLOCKS, 1:MAXROWNO]

This array is used to store the name of each constraint, the name has to be uniquely assigned otherwise program will terminate abnormally. The 1st dimension of the array denotes the particular subproblem set.

The 2nd dimension refers to a constraint of a particular subproblem set, e.g. CONSTRNAME [3,2] refers to the name for the 2nd constraint of the 3rd subproblem set.

f) FY [0:BSIZE]

This array is used to store the transformed entering column before it is pivoted into the basis inverse of subproblem set. It is set up locally in module UPDATE.

g) FY [0:MSIZE]

This array is used to transform the best proposal before it is pivoted into the master basis. It is set up locally in module PIVOTENTERCOL.

h) M [0:MSIZE, -1:MSIZE]

This 2-D array stores the basis inverse of the master program.

The 1st dimension denotes the row of the basis inverse.
The 2nd dimension denotes the column of the basis inverse and -1 column stores the values of basic variables.

i) MY [1:NUMBLOCKS, 0:MSIZE]

This array is used to store the various entering columns for the master program, formed in the respective subproblem set.
The 1st dimension refers to the particular subproblem set.
The 2nd dimension denotes the respective row in the entering column.

j) NEWB [0:SIZE, -2:2 * SIZE +1]

This array is used for reinverting the basis inverse, half of the array will become the new basis inverse at end of matrix reinversion. It is set up locally in module MATRIXREINVERSION.

k) OBJ [1:NUMBLOCKS, 1:MVAR]

This array is used to store the original coefficients of the objective function for each subproblem set (temporarily).
The 1st dimension corresponds to the subproblem set that is currently referred to.
The 2nd dimension refers to the variable associated.

l) OBJCOEFF [1:NUMOBJ, 0:NUMBLOCKS, 1:MAXCOLNO]

This array stores the coefficients of the objective function.
The 1st dimension denotes the number of the objective function that is currently referred to.

The 2nd dimension indicates the associated subproblem set.

The 3rd dimension refers to the variable which the coefficient is attached to.

m) OTINDEX [1:NUMBLOCKS, 0:OTSIZE]

The 2-D array is used to store the indices of all basic variables for various subproblem sets at optimality.

The 1st dimension refers to the particular subproblem set.

The 2nd dimension indicates the row index of the basis inverse.

n) OTS [1:NUMBLOCKS, 0:OTSIZE, -1:OTSIZE]

This array is used to store the basis inverse for various subproblem sets at optimality.

The 1st dimension is used to refer to the particular subproblem set.

The 2nd dimension refers to the row of the basis inverse.

The 3rd dimension refers to the columns of the basis inverse.

o) PROBLEMNAME [0:4]

This array is used to store the name of the user problem.

p) RCOSTF [1:NUMBLOCKS]

This array is used to compute the relative cost factor of all entering columns, formed in the various subproblem sets.

q) TEMP [-1:BSIZE]

This array is set up locally in module UPDATE for storing the leaving row before it is updated.

r) TEMP [-1:MSIZE]

This array is set up locally in module PIVOTENTERCOL to be used as temporary storages for the leaving row in master basis.

s) TEMP [0:1]

This array is used for intermediate storage purposes; it is set up locally in module ZJMINUSCJ.

t) TEMP [2:MSIZE]

This array is set up locally in module FORMENTERCOL for the use of temporary storage.

u) TEMPX [0:1]

This array is used for intermediate storage for computing purposes.

v) VARNAME [0:NUMBLOCKS, 1:MAXCOLNO]

This array is used to store the variable names assigned to the variables in each subproblem set.
The 1st dimension denotes the particular subproblem set.
The 2nd dimension denotes a variable of a particular subproblem set.

w) X [1:MAXCOLNO]

This array stores the optimal solution of each subproblem set temporarily.

x) XVALUE [1:NUMBLOCKS, 1:MVAR]

This array stores the optimal solution of various subproblem sets initially and later it is used to store the final solution of respective subproblem sets.
The 1st dimension refers to the particular subproblem set.
The 2nd dimension refers to the respective variables in that subproblem set.

y) Y [0:BSIZE]

This array is used to store the entering column formed for the basis inverse of the subproblem set.

z) ZJCJ [O:ARRAYSIZE]

This array is set up locally in module ZJMINUSCJ and is used for storing all Z_j-C_j values computed for non-basic variables.

B. Variables

i) Boolean Variables

a) FOUND

A local flag used in modules ZJMINUSCJ and MATRIXREINVERSION, it is used to denote a particular variable is/is not in the basis. It is set to FALSE initially.

b) MASTEROPTIMAL

A flag set to indicate master program has/has not reached optimality, it is set to FALSE initially.

c) NOENTERCANDIDATE

A flag set to indicate whether there are any eligible proposals to enter the master program, it is set to TRUE initially.

d) OPTIMAL

A flag set to denote the subproblem set has/has not reached optimality, it is set to a value FALSE initially.

ii) Integer Variables

a) ADVANCESOLN

This is a flag denotes whether an initial solution is/is not provided.

b) ARRAYSIZE

This parameter stores the total number of variables (including slack, surplus and artificial) of various subproblem sets.

c) BLKNO

Similar to NUMBLOCKS, set up for editing the input data.

d) BSIZE

This parameter denotes the size of basis inverse for various subproblem sets.

e) CONVEXROW

A local variable set up in module FORMENTERCOL for linking the various proposals to the corresponding convexity rows in the master basis.

f) ECOL

A local variable used in module MATRIXREINVERSION for storing the index of the entering column.

g) ENTERCOL

This parameter indicates the index of the non-basic variable which is eligible to enter the basis of the subproblem set.

h) ITERATION

A counter to keep track of the number of iterations computed for the various subproblem sets.

i) LEAVEROW

This parameter denotes the index of the leaving row in the subproblem set's basis.

j) LROW

A local variable used in module MATRIXREINVERSION for storing the index of the leaving row.

k) MAXCOLNO

This parameter is used to store the maximum number of variables among all the subproblem sets and coupling blocks.

l) MAXROWNO

This parameter is used to store the maximum number of constraints among all the subproblem sets and coupling constraints.

m) MIT

A counter for the number of master cycles.

n) MPHASE

A flag set to indicate whether the master pgoram is in PHASE I (1) or Phase II (0).

o) MSIZE

This variable indicates the size of the inverse of the master program.

p) MVAR

This parameter stores the total number of variables in each subproblem set.

q) NUMBLOCKS

This parameter indicates the number of subproblem sets in the entire problem.

r) NUMCOEFF

This parameter stores the number of real variables for each subproblem set.

s) NUMOBJ

This variable denotes the number of objective functions in the problem. By default, it has assumed to have only 1 objective function for the problem unless otherwise stated.

t) OBJNO

Similar to NUMOBJ, it is set up for editing the input data.

u) OLDINDEX

A local variable used in module UPDATE for storing the index of the basic variable which is currently leaving the basis of subproblem set.

v) OTSIZE

This variable indicates the maximum possible size of the basis inverse for subproblem sets.

w) OUTITERATION

This parameter specifies the number of master cycles to be computed before the intermediate result is written onto disk. By default it has a value 30.

x) PHASE

A flag set to denote the state of subproblem set.

0 for PHASE II.

1 for PHASE I.

y) QUITERATION

This variable denotes the number of master cycles to be computed before the program is aborted. By default, it has a value 40.

z) ROWLEAVE

A local variable used in module PIVOTENTERCOL for storing the index of leaving row in the master basis.

zz) TOLITERATION

This parameter denotes the number of iterations to be computed before a matrix reinversion is performed on the basis inverse of the subproblem set, by default, it is set to a value 20.

Other integer variables such as I, I1, I2, I3, I4, I5, I6, I7, I8, J, J1, J2, J3, J4, J5, J6, J7, J8, K, K1, K4, K6, K8, L, L1, L8, N and M1 are used as loop counter in the program.

iii) Real Variables

a) CARDID

This parameter stores the type of control card being read in. No control card has a name of more than 6 characters.

b) CONTRID

This variable stores the name of the constraint that is currently referred to, for the purpose of editing the input.

c) MBXTOL

This parameter is used to store the value of the tolerance, set for reinverting the basis inverse of a subproblem set. By default, it is set to 10 ** (-5).

d) MINRATIO

A variable declared locally in modules ZJMINUSCJ, PIVOTENTERCOL and MATRIXREINVERSION respectively for storing the minimum value of the ratio test computed so far. It is initialised to a value 10 ** (10).

e) MINZJCJ

A parameter set up locally in module MATRIXREINVERSION for computing the minimum Z_j-C_j values during the process of reinverting the matrix basis for a subproblem set.

f) PRODUCTPIVOT

This variable stores the product of pivot elements which were generated from the iterations computed so far. This is then used to compare with MBXTOL to determine the reinversion of the basis matrix.

g) RATIO

This parameter is used to store the value of the ratio test computed temporarily. It is set up locally in modules ZJMINUSCJ, PIVOTENTERCOL and MATRIXREINVERSION respectively.

h) RHSTOL

This parameter is used to store the tolerance set for the value of the basic variables before the ratio test is carried out. By default, it is set to 10 ** (-5).

i) VARID

This parameter stores the name of the variables that are currently referred to, it is set up for editing the input.

j) WOA

A parameter used to store intermediate result for computing the entering candidate for the master program, it is set up locally in FORMENTERCOL.

k) W1A

As described in WOA.

l) ZJCJ

This variable stores the Z_j-C_j values computed temporarily during the reinverting of the matrix inverse. It is set up locally in module MATRIXREINVERSION.

m) ZJCJTOL

This parameter stores the tolerance set for the Z_j-C_j rows to prevent degeneracy occuring. It is set to 10 ** (-5) by default.

C. Files

i) ADVANCEFILE

This file is set up to store the initial feasible solution to the problem specified by the user (if any).

ii) CARD

This is the input file containing information on both the control cards and data cards defining the user problem.

iii) DATAFILE

This is an option file set up for input to be read from DISK. (If required by the user).

iv) OUT1

This is a temporary output file created for storing intermediate results as soon as the problem has a feasible solution. (i.e. master program has reached end of phase I).

v) OUT2

This is a temporary output file created for storing intermediate solution of the problem whenever the master program has reached a particular iteration cycle specified by the user.

vi) OUT3

This is a temporary output file created for storing solution obtained at a stage when the master program has reached a maximum iteration cycle specified by the user.

The above iv), v) and vi) files are set up for storing intermediate solutions under various conditions, so that further processing on the same problem could be carried out at a later stage if necessary.

4. Input and Output Parameter

A. Input

The input is from punched cards and there are mainly 2 different types of cards to be dealt with:

i) Control Card

The control cards specify the type of data that follow. There are 12 types of control cards, each with a unique name of not more than 6 characters and they are each punched on a card starting from column 1.

The control cards have to appear in the order listed below:

a) 'NAME ' (Optional)

This control card denotes the data card that follows, specifies the user problem name.

b) 'NUMOBJ' (Optional)

This control card denotes the data card that follows, specifies the number of objective functions in the user problem.

c) 'NUMBLK'

This control card denotes the data card that follows, specifies the number of subproblem sets in the given problem.

d) 'NUMROW'

The data cards follow this control card indicate the number of constraints in each subproblem sets.

e) 'NUMCOL'

The data cards follow this control card indicate the number of variables in each subproblem set.

f) 'OBJCOF'

This control card denotes the data cards that follow, specifying the coefficients of the objective function(s) of the problem.

g) 'MATRIX'

The data cards following this control card indicate the coefficients of the constraints for coupling block and subproblem sets.

h) 'RHS '

This control card denotes the data cards that follow, specifying the R.H.S. values of the constraints and their status from both the coupling block and subproblem sets.

i) 'TOL ' (Optional)

This control card denotes the data card that follows, specifies the various tolerance limits set by user.

j) 'OPTION' (Optional)

The data card follows this control card specifies the various options set by user.

k) 'PRINT ' (Optional)

The data cards follow this control card specify the various print options which could be set by the user.

1) 'EOF '

This control card signifies the end of input and no data card should follow next.

ii) Data Card

All data cards must appear in the order as specified by the respective control cards. No data card should be attached when the respective optional control card is omitted. Listed below are the data format classified under the respective control cards mentioned in i):

* CARD TYPE: a

  FORMAT: <X6,5C6>

  REMARKS: Maximum problem's name of 30 characters.

* CARD TYPE: b

  FORMAT: <X3,I3>

  REMARKS: By default, it is assumed to have only 1 objective function.

* CARD TYPE: c

  FORMAT: <X3,I3>

  REMARKS: -

* CARD TYPE: d

  FORMAT: <X3,*I3>

  REMARKS: -

* CARD TYPE: e

  FORMAT: <X3,*I3>

  REMARKS: -

* CARD TYPE: f

  FORMAT: <X6,2I3,X10,C6,X2,F13.6>

REMARKS: The data cards also include the following:

1) The associated objective function no.(I3).
2) The associated subproblem set no.(I3).
3) The name of the variable corresponding to the coefficient of the objective function read (C6).
4) The respective coefficient of the objective function (F13.6).

* CARD TYPE: g

FORMAT: <X9,I3,2(X2,C6),X2,F13.6>

REMARKS: The data cards also include:

1) The associated subproblem set no.(I3).
2) The associated constraint name (C6).
3) The name of the corresponding variable.
4) The numerical value of the matrix elements.

* CARD TYPE: h

FORMAT: <X9,I3,X2,C6,X8,F13.6,X2,I2>

REMARKS: The data cards also include:

1) The associated subproblem set no.(I3).
2) The respective constraint names (C6).
3) The R.H.S. value of constraints.
4) The status of constraints.

* CARD TYPE: i

FORMAT: FREE FORMAT

REMARKS: There are 3 types of tolerance which could be specified by the user:

1) Tolerance set for pivot element before matrix reinversion, by default, it is set to a value 10 ** (-5).
2) Tolerance set for values of basic variable before ratio test, by default, it is set to a value 10 ** (-5).
3) Tolerance set for the Z_j-C_j computation, by default, it is set to a value 10 ** (-5).

* CARD TYPE: j

FORMAT: FREE FORMAT

REMARKS: There are 4 options which could be set by user:

1) Initial feasible solution is/is not (1/0) supplied, by default, it is set to a value 0.
2) Maximum number of cycles required to compute the master program, before terminating the processing, by default, it is set to 40 cycles.
3) Number of iterations required to compute the subproblem set before matrix reinversion is carried out for the basis inverse, by default, it is set to 20 iterations.

4) Number of cycles required to compute a solution for the master program before the intermediate solution is written onto disk and saved, by default, it is set to 30 cycles.

* CARD TYPE: k

FORMAT: FREE FORMAT

REMARKS: A card indicating flags for master program is read first, followed sequentially by the flags on each card for various subproblem sets.

B. Output

The output of the program consists of 3 major parts:

i) The problem name and all statistical information (such as number of objective functions, tolerance limits and other options etc. set by user are printed as soon as all input data has been read.

ii) The user can control the print out of the solutions both before and after the problem has reached its optimality, this can be done easily by setting the respective flags on the data cards of type k. By default, all these flags are set to true (or 1), indicating all relevant information are required to be printed. When the flags are reset to false (or 0) by user, the corresponding information is suppressed. There are 8 of these flags both for master program and the various subproblem sets (i.e. arrays MP [0:7] for master program and S [0:NUMBLOCKS,0:7] for subproblem sets). Listed below are the detail prescription:

FLAGS		PRINT
a)	MP [0]	Initial value of the objective functions for phase I and II.
b)	MP [1]	At each master cycle: - Status of solution (optimal/not optimal) - Cycle # - Leaving and entering vector index - Objective values for phase I and II - Product of pivot elements
c)	MP [2]	At end of pahse I for the master program: - Indices of basic variables - Values of basic variables - Objective values for phase I and II
d)	MP [3]	At end of phase I for the master program: - The matrix for master basis inverse
e)	MP [4]	At the end of phase II for the master program: (i.e. optimality) - Indices of basic variables - Values of basic variables - Objective values for phase I and II
f)	MP [5]	At the end of phase II for the master program: (i.e. optimality) - The matrix for the master basis inverse
g)	MP [6]	The subproblem set no. from which the incoming master column is generated.
h)	MP [7]	All dual prices computed for both the coupling and convexity rows in each master cycle.

i) S [K,0] Initial solution for subproblem set K:

- Indices of basic variables
- Values of basic variables
- Initial objective function value

both for phase I and II.

j) S [k,1] At each iteration (for subproblem set K):

- Status of solution (optimal/not optimal)
- Iteration #
- Leaving and entering vector index
- Objective values for phase I and II
- Product of pivot elements

k) S [K,2] At the end of phase I for subproblem set K:

- Indices of basic variables
- Values of basic variables
- Objective values for phase I and II

l) S [K,3] At the end of phase I for subproblem set K:

- The matrix for basic inverse

m) S [K,4] At the end of phase II (i.e. optimal) for subproblem set K:

- Indices of basic variables
- Values of basic variables
- Objective values for phase I and II

n) S [K,5] At the end of phase II for subproblem set K:

- The matrix for basic inverse

o) S [K,6] When reinversion takes place for subproblem set K and the value of the new product pivot element

p) S [K,7] When reinversion takes place for subproblem set K

- The basis inverse prior to reinversion

- The basis inverse after reinversion

iii) Intermediate solutions are written onto pack at the following stages:

a) The master program has reached the end of phase I and a feasible solution is obtained.

b) The master program has reached a maximum cycle number specified by the user before processing is terminated.

c) The master program has reached a particular cycle number specified by the user.

In case the program is terminated abnormally, a relevant message will be listed denoting either:

1) Error due to the input data (see Appendix D), or

2) The problem has an infeasible solution (see paragraph Generating of solutions for subproblem sets under sub-heading Program Structure), or

3) The problem has an unbounded solution (see paragraph Generating of solutions for subproblem sets under sub-heading Program Structure).

A listing of program output is shown in Appendix C.

CHAPTER V

SAMPLE PROGRAM

V <u>SAMPLE PROGRAM</u>

Consider the problem:

Maximize $Z = 4x_1 + 2X_2 + X_3 + 2X_4$

Subject to

$$
\begin{aligned}
X_1 + 4X_2 + 4X_3 + 2X_4 &= 18 \\
X_1 + 2X_2 &\leqslant 4 \\
2X_1 + X_2 &\leqslant 6 \\
X_3 + X_4 &\leqslant 4 \\
2X_3 + 4X_4 &\leqslant 10 \\
X_1,\ X_2,\ X_3,\ X_4 &\geqslant 0
\end{aligned}
$$

Assume the user wishes to set the following options:

1) Tolerance for pivot element before matrix reinversion = 0
2) Tolerance for R.H.S. values before ratio test = 0
3) Tolerance for $Z_j - C_j$ rows = 10 ** (-5)
4) Maximum number of cycles required to compute for the master program before quit = 25 (cycles).
5) Number of iterations required in the subproblem sets before matrix reinversion is carried out for the basis inverse = 20 (iterations).
6) No advance feasible solution is supplied to the problem (i.e. Flag set to 0).
7) Number of cycles required to compute before the intermediate solution is written onto pack (or disk) = 30 (cycles).
8) All relevant information required to be printed (i.e. All print option flags set to 1).

The format of the input data is shown on the page followed.

UNIVERSITY OF CANTERBURY

Data Card Sheet

PUNCH IN EVERY CARD

PROBLEM DANTZIG-WOLFE (LP) NAME C.H. OO DATE 30.10.1978

CARD COLS 1–80

CONSTANT DATA

Write in pencil with no insertions or deletions.

Distinguish clearly between 0 and O, 1 and I, 2 and Z, 5 and S.

```
NAME        maximum of 30 characters
      TEST DANTZIG-WOLFE (LP)
NUMOBJ
        I3
        1   — no. of objective function
NUMBLK
        I3
        2   — no. of subproblem sets
NUMROW
              I3   I3
        1     2    2   — no. of constraints in each subproblem sets
NUMCOL
        0     2    2   — no. of variables in each subproblem sets
OBJCOF                      variable name        F13.6
        1     1             X1VAR         -      4.00
   obj. fn. no.             X2VAR         -      2.00     coefficients
              2             Y1VAR         -      1.00     for
                                                          objective function
   subproblem               Y2VAR         -      2.00
   set no.
MATRIX             constraint name    C6
              0    CUPCON             X1VAR      1.00
                   CUPCON             X2VAR      4.00
                   CUPCON             Y1VAR      4.00     values for the
              I3   CUPCON             Y2VAR      2.00     matrix elements
              1    B11CON             X1VAR      1.00
```

Standard Graphics: 1 2 3 4 5 6 7 8 9 0 + – . / = ' * , ( ) $ &

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

UNIVERSITY OF CANTERBURY

Data Card Sheet

PUNCH IN EVERY CARD

PROBLEM DANTZIG-WOLFE (LP) NAME C.H. OO DATE 30.10.1978

CARD COLS / CONSTANT DATA

			C6	C6	F13.6		
			B11CON	X2VAR	2.00		Values for matrix elements
			B12CON	X1VAR	2.00		
		subproblem set no.	B12CON	X2VAR	1.00		
		2	B21CON	Y1VAR	1.00		
			B21CON	Y2VAR	1.00		
			B22CON	Y1VAR	2.00		
			B22CON	Y2VAR	4.00		
RHS		I3	constraint name		F13.6	I2	
		0	CUPCON		18.00	0	R.H.S. value and status of constraints
		1	B11CON		4.00	1	
			B12CON		6.00	1	0 =
		2	B21CON		4.00	1	1 ≤
			B22CON		10.00	1	−1 ≥

Write in pencil with no insertions or deletions.

Distinguish clearly between 0 and O, I and 1, 2 and Z, 5 and S.

TOL

Free Format → 0,0,0.000001, — tolerances limit set for pivot elements, R.H.S. values, Z_j-C_j rows

OPTION

Free Format → 25,20,0,15, — quit after 25 cycles, matrix for basis inverse reinverted after 20 cycles, no advance solution provided, output solution after 15 cycles.

PRINT

Free Format
1,1,1,1,1,1,1, — various print flags set for master program.
1,1,1,1,1,1,1,
1,1,1,1,1,1,1, } Print flags set for various subproblem sets.

Standard Graphics 1 2 3 4 5 6 7 8 9 0 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

UNIVERSITY OF CANTERBURY

Data Card Sheet

PUNCH IN EVERY CARD

PROBLEM DANTZIG-WOLFE (LP) NAME C.H. OO DATE 30.10.1978

CARD COLS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

CONSTANT DATA

EOF

Write in pencil with no insertions or deletions.

Distinguish clearly between 0 and Ø, I and 1, 2 and Z, 5 and S.

Standard Graphics: 1 2 3 4 5 6 7 8 9 0 + − · / = ' * , ( ) $ " &

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z S

CARD COLS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

The following points are to be noted in specifying the input data:

1) The code is written for solving a minimizing problem, thus if the problem is to maximize the objective function (as example above) the coefficients for objective function would be entered as negative values.

2) The names assigned to the various constraints and variables are to be unique and not more than 6 characters.

3) All R.H.S. values must be positive.

4) All zero matrix elements could be ignored in the input specification.

The output solution corresponding to the sample problem is shown in Appendix C.

CHAPTER VI

CONCLUSIONS

VI CONCLUSIONS

The coded program has been tested with sample problems taken from [1] and [2] and have shown to work successfully. At present, this code is being used to solve the real life forest cutting problem involving more than 500 constraints and around 1500 variables, the information generated so far has shown to be useful.

The inclusion of editing of input in the program is specifically designed for students. Allowance has been made for the user to ignore that part of processing and read the input from other media (such as disk etc.) by just replacing 1 single statement (i.e. $ SET READISK) in the code and subsequently recompile to get a new source version. In addition, the various option features added have increased the flexibility of the code and thus enable the user to have greater control in running the program.

Since the program is designed to be package-liked and very much user orientated, a detailed documentation has been included in this report for maintenance and further modification and specific use.

BIBLIOGRAPHY

1) LEON COOPER AND DAVID STEINBERG:

METHODS AND APPLICATION OF LINEAR PROGRAMMING

1974 W.B. SAUNDERS COMPANY

2) LEON S. LASDON:

OPTIMIZATION THEORY FOR LARGE SYSTEMS

1970 THE MACMILLAN COMPANY

3) JAMES L. KUESTER:

OPTIMIZATION TECHNIQUES WITH FORTRAN

1973 McGRAW-HILL BOOK COMPANY

4) STANLEY ZIONTS:

LINEAR AND INTEGER PROGRAMMING

1974 PRENTICE-HALL INC.

5) G. HADLEY:

LINEAR PROGRAMMING

1962 ADDISON-WESLEY PUBLISHING COMPANY

6) G.B. DANTZIG AND P. WOLFE:

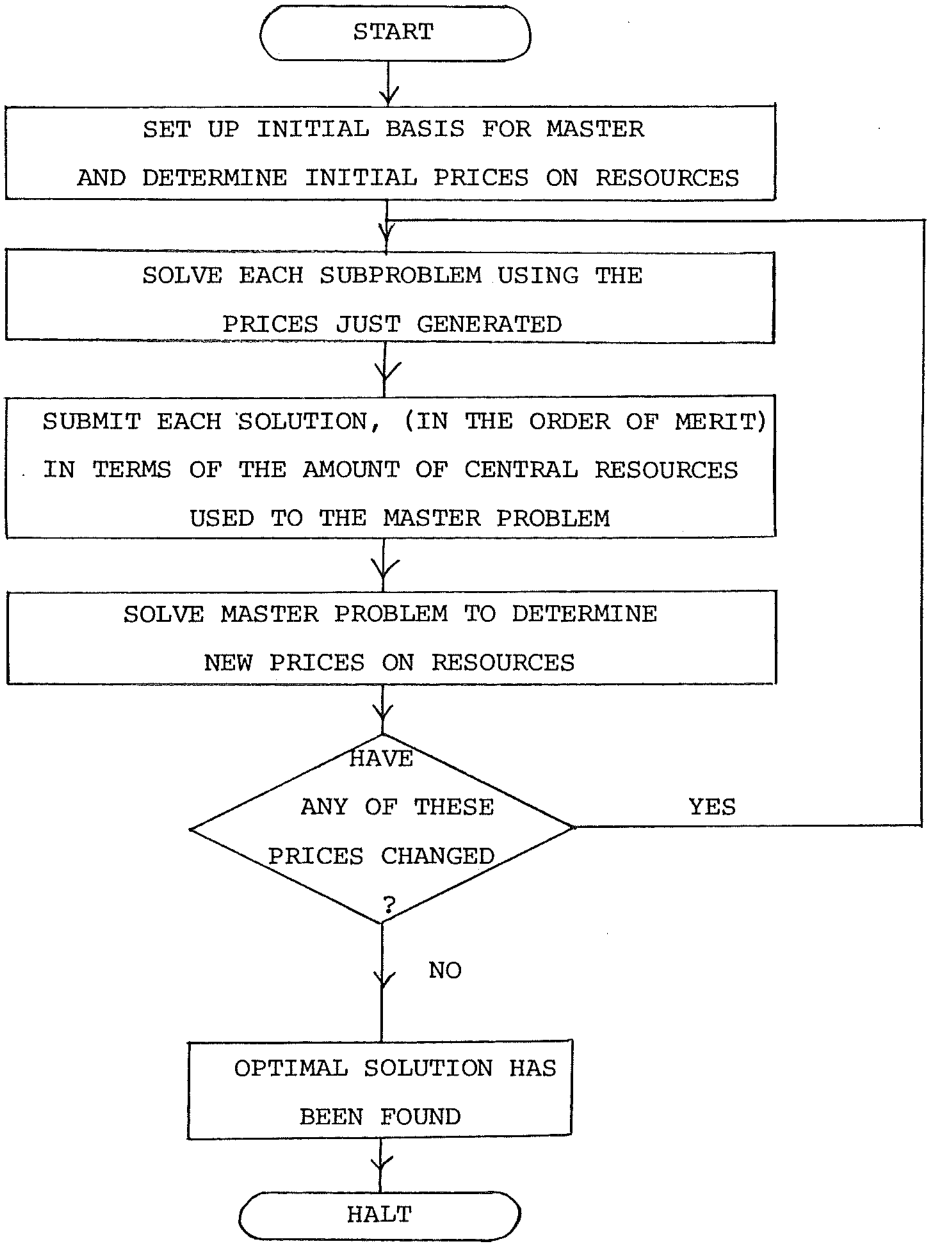
THE DECOMPOSITION PRINCIPLE FOR LINEAR PROGRAMMING

1961 ECONOMETRICA, 29 p.p. 767-778.

APPENDIX A-1

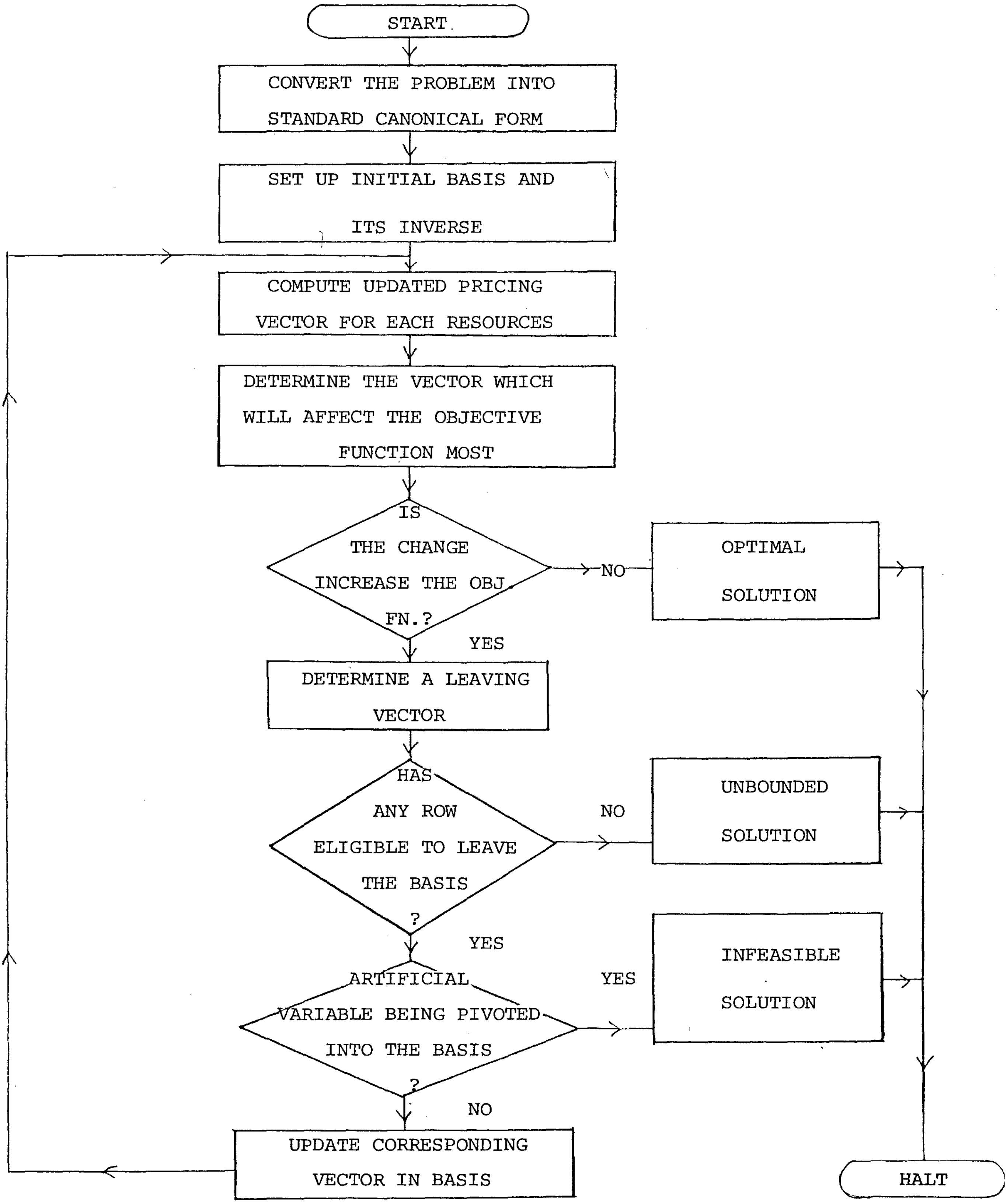
APPENDIX A-1

SIMPLIFIED FLOW CHART FOR THE DANTZIG-WOLFE DECOMPOSITION (LP) ALGORITHM



APPENDIX A-2

FLOW CHART FOR REVISED SIMPLEX METHOD



APPENDIX B

```
BURROUGHS B6700 ALGOL COMPILER, VERSION 2.9.190,    FRIDAY, 10/27/78, 05:51 PM.

                              C A R D O B J
                              = = = = = = =

        SOURCE TAPE:-  (COSC46000)CARDSOURCE ON CANDEPACK.

            NEW SYMBOLIC:  (COSC46000)NEWTAPE ON PACK.

$ SET FORMAT                                                             00001000  000:0000:0
$ RESET READISK                                                          00002000  000:0000:0
$ SET NOPRINT                                                            00003000  000:0000:0
BEGIN                                                                    00004000  000:0000:0
                                                                        B.0000 IS SEGMENT 00003
  FILE CARD(KIND=READER),                                           1    00005000  003:0000:1
                                                                              DATA IS 0005 LONG
       ADVANCEFILE(KIND=DISK),                                           00006000  003:0000:1
                                                                              DATA IS 0006 LONG
       DATAFILE(KIND=DISK,MYUSE=IN,FILETYPE=7),                          00007000  003:0000:1
                                                                              DATA IS 0006 LONG
       OUT1(KIND=DISK,MYUSE=OUT),                                        00008000  003:0000:1
                                                                              DATA IS 0005 LONG
       OUT2(KIND=DISK,MYUSE=OUT),                                        00009000  003:0000:1
                                                                              DATA IS 0005 LONG
       OUT3(KIND=DISK,MYUSE=OUT),                                        00010000  003:0000:1
                                                                              DATA IS 0005 LONG
       LINE(KIND=PRINTER);                                               00011000  003:0000:1
                                                                              DATA IS 0005 LONG
$ SET OMIT = READISK                                                     00012000  003:0000:1
$ SET LIST = NOT READISK                                                 00013000  003:0000:1
  ALPHA ARRAY PROBLEMNAME[0:4];                                          00014000  003:0000:1
$ POP OMIT LIST                                                          00015000  003:0000:1
  INTEGER ADVANCESOLN,       % FLAG TO DENOTE WHETHER INITIAL SOLN IS    00016000  003:0000:1
                             % PROVIDED.                                 00017000  003:0000:1
          F2REC,             % THE NUMBER OF RECORDS IN DATAFILE.        00018000  003:0000:1
          NUMBLOCKS,         % NUMBER OF SUBSYSTEMS.                     00019000  003:0000:1
          NUMOBJ,            % NUMBER OF OBJECTIVE FUNCTION.             00020000  003:0000:1
          OUTITERATION,      % AFTER THIS NUMBER OF ITERATIONS, THE      00021000  003:0000:1
                             % INTERMEDIATE RESULT IS WRITTEN ONTO DISK. 00022000  003:0000:1
          QUITERATION,       % AFTER THIS NUMBER OF ITERATION, THE MASTER 00023000 003:0000:1
                             % PROBLEM IS QUIT.                          00024000  003:0000:1
          TOLITERATION;      % AFTER THIS NUMBER OF ITERATION, TOLERANCE 00025000  003:0000:1
                             % OF MATRIX ELEMENTS ARE CHECKED.           00026000  003:0000:1
  ALPHA CARDID;                                                          00027000  003:0000:1
  REAL MBXTOL,               % TOLERANCE FOR SINGULARITY OF INVERSE      00028000  003:0000:1
                             % MATRIX.                                   00029000  003:0000:1
       RHSTOL,               % TOLERANCE FOR RATIO TEST.                 00030000  003:0000:1
       ZJCJTOL;              % TOLERANCE FOR ZJ-CJ ROWS.                 00031000  003:0000:1
  LABEL EOFL;                                                            00032000  003:0000:1
  DEFINE READID=READ(CARD,<C6>,CARDID)#,                                 00033000  003:0000:1
         PROGRAMABORT=GU TO EOFL#;                                       00034000  003:0000:1
$ SET OMIT = READISK                                                     00035000  003:0000:1
$ SET LIST = NOT READISK                                                 00036000  003:0000:1
  READID;                                                                00037000  003:0000:1
  IF CARDID="NAME  " THEN                                                00038000  003:0008:2
  BEGIN                                                                  00039000  003:000A:1
   READ(CARD,<X6,5C6>,PROBLEMNAME[*]);                              2    00040000  003:000A:4
   READID;                                                               00041000  003:0013:2
   WRITE(LINE,<X50,5C6,/X50,30("=")>,PROBLEMNAME[*]);                    00042000  003:001B:2
  END                                                                    00043000  003:0024:2
  ELSE WRITE(LINE,<X47,"DANZIG-WOLFE DECOMPOSITION(LP) PROBLEM",/X47,  2 00044000  003:0024:2
             38("=")>);                                                  00045000  003:0026:4
                                                                              DATA IS 0013 LONG
  IF CARDID="NUMOBJ" THEN                                                00046000  003:0029:2
  BEGIN                                                                  00047000  003:002B:1
   READ(CARD,<X6,I3>,NUMOBJ);                                       2    00048000  003:002B:4
   READID;                                                               00049000  003:0033:2
  END                                                                    00050000  003:003B:2
  ELSE NUMOBJ:=1;                                                   2    00051000  003:003B:2
$ POP OMIT LIST                                                          00052000  003:003C:3
$ SET OMIT = NOT READISK                                                 00053000     OMIT
$ POP OMIT LIST                                                          00058000  003:003C:3
```

```
          WRITE(LINE,<X50,"NUMBER OF OBJECTIVE FUNCTIONS =",I3>,NUMOBJ);                00059000   003:003C:3
                                                                                                 DATA IS 0017 LONG
       $ SET OMIT = READISK                                                              00060000   003:0043:2
       $ SET LIST = NOT READISK                                                          00061000   003:0043:2
          IF CARDID="NUMBLK" THEN                                                        00062000   003:0043:2
          BEGIN                                                                          00063000   003:0045:1
            READ(CARD,<X6,I3>,NUMBLOCKS);                                          2     00064000   003:0045:4
       $ POP OMIT LIST                                                                   00065000   003:004D:2
            WRITE(LINE,<X50,"NUMBER OF BLOCKS =",I3>,NUMBLOCKS);                         00066000   003:004D:2
       $ SET OMIT = READISK                                                              00067000   003:0054:2
       $ SET LIST = NOT READISK                                                          00068000   003:0054:2
          END                                                                            00069000   003:0054:2
          ELSE IF CARDID ¬= "NUMBLK" OR NUMBLOCKS <= 0 THEN                        2     00070000   003:0054:2
               BEGIN                                                                     00071000   003:0058:0
                 WRITE(LINE,<"***CONTROL CARD 'NUMBLK' IS EXPECTED OR THE ",       2     00072000   003:0058:3
                       "NUMBER OF BLOCKS IN THE STRUCTURE IS NOT PROPERLY ",             00073000   003:005A:2
                       "SPECIFIED***">);                                                 00074000   003:005A:2
                                                                                                 DATA IS 0017 LONG
                 PROGRAMABORT;                                                           00075000   003:005D:2
               END;                                                                      00076000   003:005D:5
       $ POP OMIT LIST                                                             2     00077000   003:005D:5
          BEGIN                                                                          00078000   003:005D:5
            BOOLEAN ARRAY MP[0:7],              % FLAGS SET TO PRINT THE           2     00079000   003:005D:5
                                                                                             B.0001 IS SEGMENT 000A
                          S[1:NUMBLOCKS,0:7];   % REQUIRED INFORMATION.                  00080000   00A:0000:1
            INTEGER ARRAY NUMCOLS[0:NUMBLOCKS],                                          00081000   00A:0003:2
                          NUMROWS[0:NUMBLOCKS];                                          00082000   00A:0006:2
            INTEGER I,                                                                   00083000   00A:0009:2
                    MAXCOLNO,                                                            00084000   00A:0009:2
                    MAXROWNO;                                                            00085000   00A:0009:2
       $ SET OMIT = READISK                                                              00086000   00A:0009:2
       $ SET LIST = NOT READISK                                                          00087000   00A:0009:2
            READID;                                                                      00088000   00A:0009:2
            IF CARDID="NUMROW" THEN READ(CARD,<X6,*I3>,NUMBLOCKS+1,NUMROWS[*])           00089000   00A:0011:2
            ELSE BEGIN                                                                   00090000   00A:001C:1
                   WRITE(LINE,<"***CONTROL CARD 'NUMROW' IS EXPECTED***">);        3     00091000   00A:0021:5
                                                                                                 DATA IS 001F LONG
                   PROGRAMABORT;                                                         00092000   00A:0026:2
                 END;                                                                    00093000   00A:0027:3
       $ POP OMIT LIST                                                             3     00094000   00A:0027:3
       $ SET OMIT = NOT READISK                                                          00095000        OMIT
       $ POP OMIT LIST                                                                   00099000   00A:0027:3
            MAXROWNO:=NUMROWS[0];                                                        00100000   00A:0027:3
            FOR I:=0 STEP 1 UNTIL NUMBLOCKS DO                                           00101000   00A:0028:4
            BEGIN                                                                        00102000   00A:002D:1
              IF MAXROWNO < NUMROWS[I] THEN MAXROWNO:=NUMROWS[I];                  3     00103000   00A:002D:1
              WRITE(LINE,<X50,"NUMBER OF ROWS IN BLOCK NO.",I3," =",I3>,I,               00104000   00A:0030:1
                                                                                                 DATA IS 000A LONG
                    NUMROWS[I]);                                                         00105000   00A:0034:4
            END;                                                                         00106000   00A:0039:2
       $ SET OMIT = READISK                                                        3     00107000   00A:0039:5
       $ SET LIST = NOT READISK                                                          00108000   00A:0039:5
            READID;                                                                      00109000   00A:0039:5
            IF CARDID="NUMCOL" THEN READ(CARD,<X6,*I3>,NUMBLOCKS+1,NUMCOLS[*])           00110000   00A:0042:2
            ELSE BEGIN                                                                   00111000   00A:004D:1
                   WRITE(LINE,<"***CONTROL CARD 'NUMCOL' IS EXPECTED***">);        3     00112000   00A:0052:5
                                                                                                 DATA IS 0014 LONG
                   PROGRAMABORT;                                                         00113000   00A:0057:2
                 END;                                                                    00114000   00A:0058:3
       $ POP OMIT LIST                                                             3     00115000   00A:0058:3
            MAXCOLNO:=NUMCOLS[0];                                                        00116000   00A:0058:3
            FOR I:=0 STEP 1 UNTIL NUMBLOCKS DO                                           00117000   00A:0059:4
            BEGIN                                                                        00118000   00A:005E:1
              IF MAXCOLNO < NUMCOLS[I] THEN MAXCOLNO:=NUMCOLS[I];                  3     00119000   00A:005E:1
              WRITE(LINE,<X50,"NUMBER OF COLUMNS IN BLOCK NO.",I3," =",I3>,I,            00120000   00A:0061:1
                                                                                                 DATA IS 000A LONG
                    NUMCOLS[I]);                                                         00121000   00A:0065:4
            END;                                                                         00122000   00A:006A:2
            BEGIN                                                                  3     00123000   00A:006A:5
       $ SET OMIT = READISK                                                        3     00124000   00A:006A:5
       $ SET LIST = NOT READISK                                                          00125000   00A:006A:5
              ALPHA ARRAY CONSTRNAME[0:NUMBLOCKS,1:MAXROWNO],                            00126000   00A:006A:5
                                                                                             B.0002 IS SEGMENT 000F
                          VARNAME[0:NUMBLOCKS,1:MAXCOLNO];                               00127000   00F:0003:5
       $ POP OMIT LIST                                                                   00128000   00F:0007:3
              INTEGER ARRAY STATUS[0:NUMBLOCKS,1:MAXROWNO],                              00129000   00F:0007:3
                            VARSTATUS[1:MAXCOLNO];                                       00130000   00F:0008:1
```

```
          REAL ARRAY A[1:NUMBLOCKS,1:MAXROWNO,1:MAXCOLNO],                       00F:000D:5
                     C[1:NUMBLOCKS,1:MAXROWNO,1:MAXCOLNO],                       00F:0011:4
                     OBJCOEFF[1:NUMOBJ,0:NUMBLOCKS,1:MAXCOLNO],                  00F:0015:3
                     RHS[0:NUMBLOCKS,1:MAXROWNO],                                00F:0019:4
                     X[1:MAXCOLNO];                                              00F:001D:2
          INTEGER J,                                                             00F:0020:0
                  K,                                                             00F:0020:0
                  L,                                                             00F:0020:0
                  BLKNO,                                                         00F:0020:0
                  MSIZE,                                                         00F:0020:0
                  MVAR,                                                          00F:0020:0
                  OBJNO,                                                         00F:0020:0
                  OTSIZE;                                                        00F:0020:0
      $ SET OMIT = READISK                                                       00F:0020:0
      $ SET LIST = NOT READISK                                                   00F:0020:0
          REAL CONSTRID,                                                         00F:0020:0
               VARID;                                                            00F:0020:0
      $ POP OMIT LIST                                                            00F:0020:0
          FOR I:=0 STEP 1 UNTIL NUMBLOCKS DO                                     00F:0020:0
          BEGIN                                                                  00F:0024:3
      $ SET OMIT = READISK                                                 4     00F:0024:3
      $ SET LIST = NOT READISK                                                   00F:0024:3
            RESIZE(CONSTRNAME[I,*],NUMROWS[I],RETAIN);                           00F:0024:3
      $ POP OMIT LIST                                                            00F:0027:1
            RESIZE(STATUS[I,*],NUMROWS[I],RETAIN);                               00F:0027:1
            RESIZE(RHS[I,*],NUMROWS[I],RETAIN);                                  00F:0029:5
      $ SET OMIT = READISK                                                       00F:002C:3
      $ SET LIST = NOT READISK                                                   00F:002C:3
            RESIZE(VARNAME[I,*],NUMCOLS[I],RETAIN);                              00F:002C:3
      $ POP OMIT LIST                                                            00F:002F:1
            FOR J:=1 STEP 1 UNTIL MAXROWNO DO                                    00F:002F:1
            IF I=0 THEN                                                          00F:0033:4
                 FOR K:=1 STEP 1 UNTIL NUMBLOCKS DO                              00F:0034:2
                 RESIZE(C[K,J,*],NUMCOLS[K],RETAIN)                              00F:0039:2
            ELSE RESIZE(A[I,J,*],NUMCOLS[I],RETAIN);                             00F:003D:0
            FOR K:=1 STEP 1 UNTIL NUMOBJ DO                                      00F:0042:3
                 RESIZE(OBJCOEFF[K,I,*],NUMCOLS[I],RETAIN);                      00F:0047:0
          END OF RESIZES;                                                        00F:004B:0
      $ SET OMIT = READISK                                                 4     00F:004B:3
      $ SET LIST = NOT READISK                                                   00F:004B:3
          READID;                                                                00F:004B:3
          IF CARDID="OBJCOF" THEN                                                00F:0053:2
          BEGIN                                                                  00F:0055:1
           WRITE(LINE,<X30,"OBJ. FN.",X2,"BLK. NO.",X2,"CONSTR. NAME",X2,  4     00F:0055:4
                 "VARIABLE NAME",X6,"VALUE",X5,"STATUS",/X30,2(8("="),X2),       00F:0057:3
                 12("="),X2,13("="),X6,5("="),X5,6("=")>);                       00F:0057:3
                                                                          DATA IS 0010 LONG
            WRITE(LINE,<X5,"INPUT DATA FOR OBJ. FN.:">);                         00F:005A:2
            FOR I:=1 STEP 1 UNTIL NUMOBJ DO                                      00F:005F:2
            BEGIN                                                                00F:0063:5
             READ(CARD [NO],<X6,2I3>,OBJNO,BLKNO);                         5     00F:0063:5
             IF OBJNO ¬= I THEN                                                  00F:006E:2
             BEGIN                                                               00F:006F:1
              WRITE(LINE,<"***INCORRECT OBJECTIVE FUNCTION NUMBER ON DATA ",  6  00F:006F:4
                    "CARD UNDER CONTROL CARD 'OBJCOF'***">);                     00F:0071:3
                                                                          DATA IS 003B LONG
              PROGRAMABORT;                                                      00F:0074:2
             END;                                                                00F:0075:3
             FOR J:=0 STEP 1 UNTIL NUMBLOCKS DO                            6     00F:0075:3
             IF NUMCOLS[J] ¬= 0 THEN                                             00F:007A:0
             BEGIN                                                               00F:007B:0
              INTEGER K;                                                   6     00F:007B:3
                                                                    B.0003 IS SEGMENT 00012
              WHILE BLKNO <= J AND K < NUMCOLS[J] DO                             012:0000:1
              BEGIN                                                              012:0002:2
               IF K > 1 THEN                                               7     012:0002:5
               IF VARID=VARNAME[J,K] THEN                                        012:0003:3
               BEGIN                                                             012:0005:5
                WRITE(LINE,<"***DUPLICATE VARIABLE NAME SPECIFIED FOR ",   8     012:0006:2
                      "OBJECTIVE FUNCTION NO.",I3,", BLOCK NO.",I3," UNDER ",    012:0008:1
                      "CONTROL CARD 'OBJCOF'***">,OBJNO,J);                      012:0008:1
                                                                          DATA IS 0012 LONG
                PROGRAMABORT;                                                    012:000E:2
               END;                                                              012:000F:3
               READ(CARD,<X22,C6,X2,F13.6>,VARNAME[J,K:=*+1],OBJCOEFF[I,J,K])  8 012:000F:3
                   ;                                                             012:001A:0
               WRITE(LINE,<X32,I3,X7,I3,X23,C6,X4,F13.6>,OBJNO,J,                012:001E:2
```

```
          VARNAME[J,K],OBJCOEFF[I,J,K]);                                    012:0024:2
  READ(CARD [NO],<X9,I3,X10,C6>,BLKNO,VARID);                               012:002D:2
 END;                                                                       012:0037:2
 IF K < NUMCOLS[J] THEN                                                  7  012:0037:5
 BEGIN                                                                      012:0039:0
  WRITE(LINE,<"***MORE DATA ARE EXPECTED FOR OBJECTIVE ",                7  012:0039:3
          "FUNCTION NO.",I3,", BLOCK NO.",I3," UNDER CONTROL CARD"          012:003B:2
          ," 'OBJCOF'***">,OBJNO,J);                                        012:003B:2
                                                                     DATA IS 0032 LONG
  PROGRAMABORT;                                                             012:0042:2
 END;                                                                       012:0043:3
 IF BLKNO <= J AND VARID ¬= "      " THEN                                7  012:0043:3
 BEGIN                                                                      012:0046:2
  WRITE(LINE,<"***TOO MANY DATA SPECIFIED FOR OBJECTIVE ",               7  012:0046:5
          "FUNCTION NO.",I3,", BLOCK NO.",I3," UNDER CONTROL ",             012:0048:4
          "CARD 'OBJCOF'***">,OBJNO,J);                                     012:0048:4
                                                                     DATA IS 001A LONG
  PROGRAMABORT;                                                             012:004F:2
 END;                                                                       012:0050:3
END OF NUMBLK;                                                           7  012:0050:3
                                                             B.0003(012) IS 005D LONG
IF BLKNO > NUMBLOCKS THEN                                                6  00F:007C:4
BEGIN                                                                       00F:007D:3
 WRITE(LINE,<"***TOO MANY BLOCKS SPECIFIED FOR OBJECTIVE ",              6  00F:007E:0
         "FUNCTION NO.",I3," UNDER CONTROL CARD 'OBJCOF'***">,              00F:007F:5
                                                                     DATA IS 001A LONG
         OBJNO);                                                            00F:0080:4
 PROGRAMABORT;                                                              00F:0085:2
END;                                                                        00F:0086:3
END OF NUMOBJ;                                                           6  00F:0086:3
END                                                                      5  00F:0087:0
ELSE BEGIN                                                               4  00F:0087:0
       WRITE(LINE,<"***CONTROL CARD 'OBJCOF' IS EXPECTED***">);          4  00F:0087:3
                                                                     DATA IS 0016 LONG
       PROGRAMABORT;                                                        00F:008C:2
      END;                                                                  00F:008D:3
READID;                                                                  4  00F:008D:3
IF CARDID="MATRIX" THEN                                                     00F:0095:2
BEGIN                                                                       00F:0097:1
 WRITE(LINE,<X5,"INPUT DATA FOR MATRIX:">);                              4  00F:0097:4
 FOR I:=0 STEP 1 UNTIL NUMBLOCKS DO                                         00F:009C:2
 BEGIN                                                                      00F:00A0:5
  READ(CARD [NO],<X9,I3,X2,C6>,BLKNO,CONSTRID);                          5  00F:00A0:5
  IF BLKNO ¬= I THEN                                                        00F:00AB:2
  BEGIN                                                                     00F:00AC:1
   WRITE(LINE,<"***INCORRECT BLOCK NUMBER ON DATA CARD UNDER ",          6  00F:00AC:4
          "CONTROL CARD 'MATRIX'***">);                                     00F:00AE:3
                                                                     DATA IS 001B LONG
  PROGRAMABORT;                                                             00F:0081:2
  END;                                                                      00F:0082:3
  FOR J:=1 STEP 1 UNTIL NUMROWS[I] DO                                    6  00F:0082:3
  BEGIN                                                                     00F:0087:2
   INTEGER K;                                                            6  00F:0087:2
                                                          B.0004 IS SEGMENT 00019
   IF J > 1 THEN                                                            019:0000:1
   IF CONSTRID=CONSTRNAME[I,J-1] THEN                                       019:0000:5
   BEGIN                                                                    019:0003:3
    WRITE(LINE,<"***TOO MANY DATA OR WRONG VARIABLE NAME ",              7  019:0004:0
           "SPECIFIED FOR CONSTRAINT ",C6," IN BLOCK NO.",I3,               019:0005:5
           " UNDER CONTROL CARD 'MATRIX'***">,CONSTRID,I);                  019:0005:5
                                                                     DATA IS 0010 LONG
    PROGRAMABORT;                                                           019:000C:2
   END;                                                                     019:000D:3
   READ(CARD [NO],<X9,I3,2(X2,C6)>,BLKNO,CONSTRNAME[I,J],VARID);         7  019:000D:3
   IF BLKNO > I THEN                                                        019:0018:2
   BEGIN                                                                    019:001C:1
    WRITE(LINE,<"***MORE CONSTRAINTS ARE EXPECTED FOR BLOCK NO.",        7  019:001C:4
          I3," UNDER CONTROL CARD 'MATRIX'***">,I);                         019:001E:3
                                                                     DATA IS 0024 LONG
    PROGRAMABORT;                                                           019:0023:2
   END;                                                                     019:0024:3
   WHILE CONSTRID=CONSTRNAME[I,J] DO                                     7  019:0024:3
   BEGIN                                                                    019:0026:2
    IF I¬=0 AND K>NUMCOLS[I] OR L=NUMBLOCKS AND K=NUMCOLS[L] THEN        7  019:0026:5
    BEGIN                                                                   019:002B:1
     WRITE(LINE,<"***TOO MANY DATA OR WRONG VARIABLE NAME ",             8  019:002B:4
            "SPECIFIED FOR CONSTRAINT ",C6," IN BLOCK NO.",I3,              019:002D:3
```

```
          " UNDER CONTROL CARD 'MATRIX'***">,CONSTRID,I);                          019:002D:3
                                                                            DATA IS 0013 LONG
    PROGRAMABORT;                                                                  019:0034:2
   END;                                                                            019:0035:3
   IF I=0 THEN                                                           8         019:0035:3
   BEGIN                                                                           019:0036:1
    IF K >= NUMCOLS[L] THEN                                              8         019:0036:4
    BEGIN                                                                          019:0037:5
     L:=*+1;                                                             9         019:0038:2
     K:=0;                                                                         019:0039:4
    END;                                                                           019:003A:2
    READ(CARD,<X30,F13.6>,C[L,J,K:=*+1]);                                9         019:003A:2
    WRITE(LINE,<X42,I3,X8,C6,X9,C6,X4,F13.6>,I,CONSTRID,VARID,                     019:0046:2
          C[L,J,K]);                                                               019:004D:4
    READ(CARD [NO],<X14,C6,X2,C6>,CONSTRID,VARID);                                 019:0054:2
   END                                                                             019:005E:2
   ELSE IF VARID=VARNAME[I,K:=*+1] THEN                                  8         019:005E:2
   BEGIN                                                                           019:0062:0
    READ(CARD,<X30,F13.6>,A[I,J,K]);                                     8         019:0062:3
    WRITE(LINE,<X42,I3,X8,C6,X9,C6,X4,F13.6>,I,CONSTRID,VARID,                     019:006D:2
          A[I,J,K]);                                                               019:0074:4
    READ(CARD [NO],<X14,C6,X2,C6>,CONSTRID,VARID);                                 019:0078:2
   END;                                                                            019:0085:2
  END OF CONSTRS;                                                        8         019:0085:2
 END OF NUMROW;                                                          7         019:0085:5
                                                                      B.0004(019) IS 009B LONG
END OF NUMBLK;                                                           6         00F:0088:3
END                                                                      5         00F:0089:0
ELSE BEGIN                                                               4         00F:0089:0
  WRITE(LINE,<"***CONTROL CARD 'MATRIX' IS EXPECTED***">);               4         00F:0089:3
                                                                            DATA IS 0045 LONG
  PROGRAMABORT;                                                                    00F:008E:2
 END;                                                                              00F:008F:3
READID;                                                                  4         00F:008F:3
IF CARDID="RHS   " THEN                                                            00F:00C7:2
BEGIN                                                                              00F:00C9:1
 WRITE(LINE,<X5,"INPUT FOR RHS OF CONSTR.:">);                           4         00F:00C9:4
 FOR I:=0 STEP 1 UNTIL NUMBLOCKS DO                                                00F:00CE:2
 BEGIN                                                                             00F:00D2:5
  READ(CARD [NO],<X9,I3,X2,C6>,BLKNO,CONSTRID);                          5         00F:00D2:5
  IF BLKNO ¬= I THEN                                                               00F:00DD:2
  BEGIN                                                                            00F:00DE:1
   WRITE(LINE,<"***INCORRECT BLOCK NUMBER ON DATA CARD UNDER ",          6         00F:00DE:4
         "CONTROL CARD 'RHS    '***">);                                            00F:00E0:3
                                                                            DATA IS 001C LONG
  PROGRAMABORT;                                                                    00F:00E3:2
  END;                                                                             00F:00E4:3
  FOR J:=1 STEP 1 UNTIL NUMROWS[I] DO                                    6         00F:00E4:3
  BEGIN                                                                            00F:00E9:2
   IF J > 1 THEN                                                         6         00F:00E9:2
   IF CONSTRID=CONSTRNAME[I,J-1] THEN                                              00F:00EA:0
   BEGIN                                                                           00F:00EC:4
    WRITE(LINE,<"***DUPLICATE CONSTRAINT NAME USED IN BLOCK NO.",        7         00F:00ED:1
          I3," UNDER CONTROL CARD 'RHS    '***">,I);                               00F:00EF:0
                                                                            DATA IS 0010 LONG
    PROGRAMABORT;                                                                  00F:00F4:2
   END;                                                                            00F:00F5:3
   IF BLKNO > I THEN                                                     7         00F:00F5:3
   BEGIN                                                                           00F:00F6:2
    WRITE(LINE,<"***MORE CONSTRAINTS ARE EXPECTED FOR BLOCK NO.",        7         00F:00F6:5
          I3," UNDER CONTROL CARD 'RHS    '***">,I);                               00F:00F8:4
                                                                            DATA IS 0013 LONG
   END;                                                                            00F:00FE:2
   IF CONSTRID=CONSTRNAME[I,J] THEN                                      7         00F:00FE:2
   BEGIN                                                                           00F:0100:1
    READ(CARD,<X30,F13.6,X2,I2>,RHS[I,J],STATUS[I,J]);                   7         00F:0100:4
    WRITE(LINE,<X42,I3,X8,C6,X19,F13.6,X4,I2>,I,CONSTRID,                          00F:010E:2
          RHS[I,J],STATUS[I,J]);                                                   00F:0114:2
    IF STATUS[I,J] > 1 OR STATUS[I,J] < -1 THEN                                    00F:011D:2
    BEGIN                                                                          00F:0121:0
     WRITE(LINE,<"***INVALID STATUS SPECIFIED FOR CONSTRAINT ",C6,       8         00F:0121:3
           " IN BLOCK NO.",I3," UNDER CONTROL CARD 'RHS    '***">,                 00F:0123:2
                                                                            DATA IS 0023 LONG
           CONSTRID,I);                                                            00F:0124:1
     PROGRAMABORT;                                                                 00F:012A:2
    END;                                                                           00F:012B:3
    READ(CARD [NO],<X9,I3,X2,C6>,BLKNO,CONSTRID);                        8         00F:012B:3
```

```
                END
                ELSE BEGIN
                     WRITE(LINE,<"***WRONG CONSTRAINT NAME USED FOR BLOCK NO."
                          ,I3," UNDER CONTROL CARD 'RHS    '***">,I);

                    PROGRAMABORT;
                   END;
              END OF NUMROW;
              IF CONSTRID=CONSTRNAME[I,NUMROWS[I]] THEN
              BEGIN
               WRITE(LINE,<"***TOO MANY CONSTRAINTS SPECIFIED FOR BLOCK NO.",
                     I3," UNDER CONTROL CARD 'RHS    '***">,I);

               PROGRAMABORT;
              END;
             END OF NUMBLK;
            END
            ELSE BEGIN
                  WRITE(LINE,<"***CONTROL CARD 'RHS    ' IS EXPECTED***">);

                  PROGRAMABORT;
                 END;
            READ(CARD [NO],<C6>,CARDID);
            IF CARDID="TOL    " THEN
            BEGIN
             READID;
             READ(CARD,/,MBXTOL,RHSTOL,ZJCJTOL);
            END
            ELSE MBXTOL:=RHSTOL:=ZJCJTOL:=0.00001;
$ POP OMIT LIST
$ SET OMIT = NOT READISK
$ POP OMIT LIST
            WRITE(LINE [SKIP 1]);
            WRITE(LINE,<X5,"TOLERANCE FOR PIVOT ELEMENT BEFORE MATRIX ",
                  "REINVERSION =",T70,E20.6,/X5,"TOLERANCE FOR R.H.S. OF ",
                  "CONSTRAINT BEFORE RATIO TEST =",T70,E20.6,/X5,"TOLERANCE ",
                  "FOR ZJ-CJ ROWS =",T70,E20.6>,MBXTOL,RHSTOL,ZJCJTOL);
$ SET OMIT = READISK
$ SET LIST = NOT READISK
            READ(CARD [NO], <C6>, CARDID);
            IF CARDID="OPTION" THEN
            BEGIN
             READID;
             READ(CARD,/,QUITERATION,TOLITERATION,ADVANCESOLN,OUTITERATION);
            END
            ELSE BEGIN
                  ADVANCESOLN:=0;
                  OUTITERATION:=30;
                  QUITERATION:=40;
                  TOLITERATION:=20;
                 END;
$ POP OMIT LIST
            WRITE(LINE,<X5,"FLAG DENOTING THE SUPPLY OF ADVANCED SOLUTION =",
                  T77,I3,/X5,"MAXIMUM NUMBER OF ITERATIONS REQUIRED TO ",
                  "COMPUTE THE MASTER PROBLEM =",T77,I3,/X5,"NUMBER OF ",
                  "ITERATIONS REQUIRED TO COMPUTE BEFORE MATRIX REINVERSION ="
                  ,T77,I3>,ADVANCESOLN,QUITERATION,TOLITERATION);

            WRITE(LINE,<X5,"OUTPUT INTERMEDIATE RESULTS ONTO DISK AFTER THIS "
                  ,"NUMBER OF ITERATIONS =",T77,I3>,OUTITERATION);

            READ(CARD [NO],<C6>,CARDID);
            IF CARDID="PRINT " THEN
            BEGIN
             READID;
             READ(CARD,/,MP[*]);
             FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO READ(CARD,/,S[I,*]);
            END
            ELSE BEGIN
                  FOR I:=0 STEP 1 UNTIL 7 DO MP[I]:=TRUE;
                  FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO
                  FOR J:=0 STEP 1 UNTIL 7 DO S[I,J]:=TRUE;
                 END;
            WRITE(LINE,<X5,"FLAGS FOR MASTER:",T40,8I4>,MP[*]);
            FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO
            WRITE(LINE,<X5,"FLAGS FOR SUBSYSTEMS #",I3,":",T40,8I4>,I,S[I,*]);
```

```
$ SET OMIT = READISK
$ SET LIST = NOT READISK
    READID;
    IF CARDID ¬= "EOF    " THEN
    BEGIN
     WRITE(LINE,<"***CONTROL CARD 'EOF    ' IS EXPECTED***">);

     PROGRAMABORT;
    END;
    WRITE(LINE,<X5,"@@@INPUT DATA READ IN CORRECTLY, START ",
          "PROCESSING!">);

$ POP OMIT LIST
$ SET OMIT = NOT READISK
$ POP OMIT LIST
    MSIZE:=NUMROWS[0]+NUMBLOCKS+1;
    MVAR:=2 * MSIZE;
    OTSIZE:=MAXROWNO + 1 ;
    FOR L:=1 STEP 1 UNTIL NUMOBJ DO
    BEGIN
     BOOLEAN CHECKOPTIMAL,

             MASTEROPTIMAL,
             NOENTERCANDIDATE,
             OPTIMAL;
     INTEGER ARRAYSIZE,
             BSIZE,
             ENTERCOL,
             I4,
             J4,
             K4,
             L4,
             LEAVEROW,
             MIT,
             MPHASE,
             N,
             PHASE;
     REAL    PRODUCTPIVOT;
     INTEGER ARRAY NUMVARS[1:NUMBLOCKS],
                   WEIGHT[2:MSIZE],
                   XSIZE[1:NUMBLOCKS];
     REAL ARRAY BIGX[2:MSIZE,1:MVAR],
                M[0:MSIZE,-1:MSIZE],
                MY[1:NUMBLOCKS,0:MSIZE],
                OBJ[1:NUMBLOCKS,1:MVAR],
                OTINDEX[1:NUMBLOCKS,0:OTSIZE],
                OTS[1:NUMBLOCKS,0:OTSIZE,-1:OTSIZE],
                RCOSTF[1:NUMBLOCKS],
                TEMPX[0:1],
                XVALUE[1:NUMBLOCKS,1:MVAR];
     DEFINE  UPTO = STEP 1 UNTIL#;
     SWITCH FILE OUTFILE:= OUT1, OUT2, OUT3;
      PROCEDURE WRITETODISK(FILEID);
      VALUE FILEID;
      INTEGER FILEID;
      BEGIN
   % TO WRITE THE SOLUTIONS ONTO THE OUTFILE.
       WRITE(OUTFILE[FILEID],*,NUMVARS[*]);
       FOR I4:=1 UPTO NUMBLOCKS DO
       BEGIN
        J4:= NUMROWS[I4] + 1;
    % WRITE THE OPTIMAL TABLEAU FOR EACH SUBSYSTEM.
        FOR K4:= 0 UPTO J4 DO
        WRITE(OUTFILE[FILEID],*,FOR L4:=-1 UPTO J4 DO OTS[I4,K4,L4]);
    % WRITE THE INDICES OF VARIABLES IN THE BASIS OF EACH SUBSYSTEM
        WRITE(OUTFILE[FILEID],*,FOR L4:=0 UPTO J4 DO OTINDEX[I4,L4]);
    % WRITE OUT THE OPTIMAL SOLUTIONS FOR EACH SUBSYSTEM.
        WRITE(OUTFILE[FILEID],*,FOR L4:=1 UPTO NUMVARS[I4] DO
              XVALUE[I4,L4]);
       END;
   % WRITE THE BASIS FOR MASTER PROBLEM.
       FOR I4:=0 UPTO MSIZE DO
       WRITE(OUTFILE[FILEID],*,FOR L4:=-1 UPTO MSIZE DO M[I4,L4]);
       WRITE(OUTFILE[FILEID],*,WEIGHT[*]);
       FOR I4:=2 UPTO MSIZE DO
       BEGIN
```

```
     DATA IS 0023 LONG
  00416000   00F:0208:5
  00417000   00F:0208:5
  00418000   00F:0208:5
  00419000   00F:0214:2
  00420000   00F:0216:1
4 00421000   00F:0216:4
     DATA IS 0010 LONG
  00422000   00F:021B:2
  00423000   00F:021C:3
4 00424000   00F:021C:3
  00425000   00F:021E:2
     DATA IS 000A LONG
  00426000   00F:0221:2
  00427000     OMIT
  00468000   00F:0221:2
  00469000   00F:0221:2
  00470000   00F:0223:2
  00471000   00F:0224:5
  00472000   00F:0226:1
  00473000   00F:022A:4
4 00474000   00F:022A:4
 B.0005 IS SEGMENT 002C
  00475000   02C:0000:1
  00476000   02C:0000:1
  00477000   02C:0000:1
  00478000   02C:0000:1
  00479000   02C:0000:1
  00480000   02C:0000:1
  00481000   02C:0000:1
  00482000   02C:0000:1
  00483000   02C:0000:1
  00484000   02C:0000:1
  00485000   02C:0000:1
  00486000   02C:0000:1
  00487000   02C:0000:1
  00488000   02C:0000:1
  00489000   02C:0000:1
  00490000   02C:0000:1
  00491000   02C:0000:1
  00492000   02C:0002:5
  00493000   02C:0005:5
  00494000   02C:0008:3
  00495000   02C:000C:1
  00496000   02C:0010:2
  00497000   02C:0014:0
  00498000   02C:0017:2
  00499000   02C:001B:0
  00500000   02C:001F:4
  00501000   02C:0022:2
  00502000   02C:0022:2
  00503000   02C:0025:4
  00504000   02C:0025:4
  00505000   02C:0029:5
  00506000   02C:0029:5
  00507000   02C:0029:5
  00508000   02C:0029:5
  00509000   02C:0029:5
  00510000   02C:0029:5
5 00511000   02C:0035:2
  00512000   02C:0039:5
6 00513000   02C:0039:5
  00514000   02C:003B:3
  00515000   02C:003B:3
  00516000   02C:0040:0
  00517000   02C:0051:5
  00518000   02C:0051:5
  00519000   02C:0062:2
  00520000   02C:0062:2
  00521000   02C:0068:1
  00522000   02C:0073:2
6 00523000   02C:0073:5
  00524000   02C:0073:5
  00525000   02C:0078:2
  00526000   02C:0089:5
  00527000   02C:0094:2
  00528000   02C:0099:0
```

```
    J4:= -(WEIGHT[I4] MOD 100);                                   6   00529000   02C:0099:0
   IF J4 > 0  AND  J4 <= NUMBLOCKS THEN                               00530000   02C:009B:3
    WRITE(OUTFILE[FILEID],*,FOR L4:=1 UPTO NUMVARS[J4] DO             00531000   02C:009C:5
           BIGX[I4,L4]);                                              00532000   02C:00A6:1
   END;                                                               00533000   02C:00AF:2
   LOCK(OUTFILE[FILEID]);                                         6   00534000   02C:00AF:5
  END OF WRITETODISK;                                                 00535000   02C:00B1:5
```

```
PROCEDURE ADDSLASURART(I);                                                       5    00536000   02C:00C0:5
VALUE I;                                                                              00537000   02C:00C0:5
INTEGER I;                                                                            00538000   02C:00C0:5
BEGIN                                                                                 00539000   02C:00C0:5
 INTEGER I1,                                                                          00540000   02C:00C0:5
                                                                                ADDSLASURART IS SEGMENT 002E
         J1,                                                                     5    00541000   02E:0000:1
         K1;                                                                          00542000   02E:0000:1
 RESIZE(X,ARRAYSIZE,RETAIN);                                                          00543000   02E:0000:1
 RESIZE(VARSTATUS,ARRAYSIZE,RETAIN);                                                  00544000   02E:0002:1
 RESIZE(OBJCOEFF[L,I,*],ARRAYSIZE,RETAIN);                                            00545000   02E:0004:1
 I1:=NUMCOLS[I];                                                                      00546000   02E:0007:2
 FOR J1:=1 STEP 1 UNTIL NUMROWS[I] DO                                                 00547000   02E:0008:4
 BEGIN                                                                                00548000   02E:000D:3
  RESIZE(A[I,J1,*],ARRAYSIZE,RETAIN);                                            6    00549000   02E:000D:3
  FOR K1:=NUMCOLS[I]+1 STEP 1 UNTIL ARRAYSIZE DO                                      00550000   02E:0011:0
      A[I,J1,K1]:=0;                                                                  00551000   02E:0015:3
  CASE STATUS[I,J1]+2 OF                                                              00552000   02E:0019:1
  BEGIN                                                                               00553000   02E:001B:0
   1:X[I1:=*+1]:=0;              VARSTATUS[I1]:=5; A[I,J1,I1]:=-1;               7    00554000   02E:001B:0
     X[I1:=*+1]:=RHS[I,J1];      VARSTATUS[I1]:=8; A[I,J1,I1]:=1;                     00555000   02E:0025:1
   2:X[I1:=*+1]:=RHS[I,J1];      VARSTATUS[I1]:=8; A[I,J1,I1]:=1;                     00556000   02E:002D:4
   3:X[I1:=*+1]:=RHS[I,J1];      VARSTATUS[I1]:=4; A[I,J1,I1]:=1;                     00557000   02E:0036:4
  END;                                                                                00558000   02E:003F:4
 END;                                                                            7    00559000   02E:0042:0
 FOR K1:=1 STEP 1 UNTIL NUMROWS[I] DO                                            6    00560000   02E:0042:3
     RESIZE(A[I,K1,*],I1,RETAIN);                                                     00561000   02E:0047:2
 ARRAYSIZE:=I1;                                                                       00562000   02E:004B:2
END OF ADDSLASURART;                                                                  00563000   02E:004C:2
                                                                               ADDSLASURART(02E) IS 004E LONG


PROCEDURE SETUPBASIS(ARRAYSIZE,B,BINDEX,I);                                      5    00564000   02C:00C0:5
VALUE ARRAYSIZE,                                                                      00565000   02C:00C0:5
      I;                                                                              00566000   02C:00C0:5
INTEGER ARRAYSIZE,                                                                    00567000   02C:00C0:5
        I;                                                                            00568000   02C:00C0:5
INTEGER ARRAY BINDEX[0];                                                              00569000   02C:00C0:5
REAL ARRAY B[0,-1];                                                                   00570000   02C:00C0:5
BEGIN                                                                                 00571000   02C:00C0:5
 INTEGER I1,                                                                          00572000   02C:00C0:5
                                                                                 SETUPBASIS IS SEGMENT 002F
         J1,                                                                     5    00573000   02F:0000:1
         K1;                                                                          00574000   02F:0000:1
% PHASE I AND PHASE II OBJ FN IN THE BASIS.                                           00575000   02F:0000:1
 B[0,0]:=B[1,1]:=1;                                                                   00576000   02F:0000:1
 B[0,-1]:=B[1,-1]:=0;                                                                 00577000   02F:0002:5
 BINDEX[0]:=0;                                                                        00578000   02F:0005:4
 BINDEX[1]:=-1;                                                                       00579000   02F:0006:4
 J1:=1;                                                                               00580000   02F:0007:5
 FOR I1:=1 STEP 1 UNTIL ARRAYSIZE DO                                                  00581000   02F:0008:3
% SLACK VARIABLE IN THE BASIS.                                                        00582000   02F:000D:0
 IF VARSTATUS[I1]=4 THEN                                                              00583000   02F:000D:0
 BEGIN                                                                                00584000   02F:000E:3
  B[J1:=*+1,J1]:=1;                                                              6    00585000   02F:000F:0
  BINDEX[J1]:=I1;                                                                     00586000   02F:0012:0
  X[I1]:=B[J1,-1]:=RHS[I,K1:=*+1];                                                    00587000   02F:0013:3
  END                                                                                 00588000   02F:0019:0
 ELSE IF VARSTATUS[I1]=8 THEN       % ART. VAR. IN THE BASIS.                    6    00589000   02F:0019:0
      BEGIN                                                                           00590000   02F:001B:0
       B[1,J1:=*+1]:=-1;                                                         6    00591000   02F:001B:3
       B[J1,J1]:=1;                                                                   00592000   02F:001E:3
       BINDEX[J1]:=I1;                                                                00593000   02F:0020:3
       X[I1]:=B[J1,-1]:=RHS[I,K1:=*+1];                                               00594000   02F:0022:0
      END;                                                                            00595000   02F:0027:3
END OF SETUPBASIS;                                                               6    00596000   02F:0028:0
                                                                                 SETUPBASIS(02F) IS 002A LONG
```

```
PROCEDURE ZJMINUSCJ(ARRAYSIZE,B,BINDEX,BSIZE,I,ITERATION,PHASE,   5   02C:00C0:5
                    Y);                                               02C:00C0:5
VALUE ARRAYSIZE,                                                      02C:00C0:5
      BSIZE,                                                          02C:00C0:5
      I,                                                              02C:00C0:5
      PHASE;                                                          02C:00C0:5
INTEGER ARRAYSIZE,                                                    02C:00C0:5
        BSIZE,                                                        02C:00C0:5
        I,                                                            02C:00C0:5
        ITERATION,                                                    02C:00C0:5
        PHASE;                                                        02C:00C0:5
INTEGER ARRAY BINDEX[0];                                              02C:00C0:5
REAL ARRAY B[0,-1],                                                   02C:00C0:5
           Y[0];                                                      02C:00C0:5
BEGIN                                                                 02C:00C0:5
 BOOLEAN FOUND;                                                       02C:00C0:5
                                                      ZJMINUSCJ IS SEGMENT 00030
 INTEGER I1,                                                      5   030:0000:1
         J1,                                                          030:0000:1
         K1,                                                          030:0000:1
         L1,                                                          030:0000:1
         M1;                                                          030:0000:1
 REAL ARRAY TEMP[0:1],                                                030:0000:1
            ZJCJ[0:ARRAYSIZE];                                        030:0000:1
 OPTIMAL:=FALSE;                                                      030:0003:1
% COMPUTE VALUE OF OBJ FN FOR PHASE I AND II.                         030:0003:5
 IF PHASE=1 AND ITERATION=0 THEN                                      030:0003:5
 BEGIN                                                                030:0005:2
  FOR M1:=0,1 DO                                                  6   030:0005:5
  FOR L1:=0 STEP 1 UNTIL BSIZE DO TEMP[M1]:=*+B[M1,L1]*B[L1,-1]       030:0009:4
        ;                                                             030:0012:1
  FOR M1:=0,1 DO  B[M1,-1]:=TEMP[M1];                                 030:0015:3
 END;                                                                 030:001E:3
 ZJCJ[0]:=-ZJCJTOL;                                               6   030:001E:3
 ENTERCOL:=LEAVEROW:=0;                                               030:001F:5
 IF PHASE=1 AND B[1,-1]=0 THEN PHASE:=0;                              030:0021:1
 FOR I1:=1 STEP 1 UNTIL ARRAYSIZE DO                                  030:0024:2
 BEGIN                                                                030:0028:5
  FOUND:=FALSE;                                                   6   030:0028:5
  FOR J1:=2 STEP 1 WHILE(J1 <= BSIZE AND NOT FOUND) DO                030:0029:3
  IF BINDEX[J1]=I1 THEN FOUND:=TRUE;                                  030:002D:5
  IF FOUND THEN                                                       030:0030:4
  IF PHASE=1 AND ITERATION=0 THEN                                     030:0031:0
  BEGIN                                                               030:0033:0
   J1:=*-1;                                                       7   030:0033:3
   X[I1]:=0;                                                          030:0034:5
   FOR L1:=0 STEP 1 UNTIL BSIZE DO                                    030:0036:2
   X[I1]:=*+B[J1,L1] * B[L1,-1];                                      030:003A:5
   B[J1,-1]:=X[I1];                                                   030:003F:4
  END                                                                 030:0042:1
  ELSE X[I1]:=B[J1:=*-1,-1]                                       7   030:0042:1
  ELSE IF PHASE = 1  OR  VARSTATUS[I1] ¬= 7 THEN                      030:0046:2
       BEGIN                                                          030:0049:1
        ZJCJ[I1]:=B[PHASE,0]*(IF I1 <= NUMCOLS[I] THEN            7   030:0049:4
                  OBJCOEFF[L,I,I1] ELSE 0) + B[PHASE,1] *             030:004C:3
                  (IF VARSTATUS[I1]=8 THEN 1 ELSE 0);                 030:0051:1
        FOR L1:=2 STEP 1 UNTIL BSIZE DO                               030:0054:1
        ZJCJ[I1]:=*+B[PHASE,L1]*A[I,L1-1,I1];                         030:0058:5
        IF ZJCJ[I1] < ZJCJ[0] THEN                                    030:005F:0
        BEGIN                                                         030:0060:2
         ZJCJ[0]:=ZJCJ[I1];                                       8   030:0060:5
         ENTERCOL:=I1;                                                030:0062:2
        END;                                                          030:0063:2
        END OF ZJCJ;                                              8   030:0063:2
 END;                                                             7   030:0063:2
 IF ENTERCOL=0 AND ZJCJ[0]=-ZJCJTOL THEN OPTIMAL:=TRUE;           6   030:0063:5
 IF NOT OPTIMAL THEN                                                  030:0067:0
 BEGIN                                                                030:0067:2
  INTEGER I2,                                                     6   030:0067:5
                                                      B.0006 IS SEGMENT 00031
          J2;                                                         031:0000:1
  REAL MINRATIO,                                                      031:0000:1
       RATIO;                                                         031:0000:1
  FOR I2:=0 STEP 1 UNTIL BSIZE DO                                     031:0000:1
  BEGIN                                                               031:0004:4
   Y[I2]:=B[I2,0] * (IF ENTERCOL <= NUMCOLS[I] THEN               7   031:0004:4
          OBJCOEFF[L,I,ENTERCOL] ELSE 0) + B[I2,1] *                  031:0007:3
```

```
          (IF VARSTATUS[ENTERCOL] = 8 THEN 1 ELSE 0);              00672000   031:000C:1
   FOR J2:=2 STEP 1 UNTIL BSIZE DO                                 00673000   031:000F:1
   Y[I2]:=*+B[I2,J2]*A[I,J2-1,ENTERCOL];                           00674000   031:0013:5
  END;                                                             00675000   031:001A:0
% COMPUTE RATIO.                                                 7 00676000   031:001A:3
  LEAVEROW:=0;                                                     00677000   031:001A:3
  MINRATIO:= 10**10;                                               00678000   031:001B:1
  FOR I2:=2 STEP 1 UNTIL BSIZE DO                                  00679000   031:001D:3
  BEGIN                                                            00680000   031:0022:1
  IF B[I2,-1] < RHSTOL THEN B[I2,-1]:= RHSTOL * I2;              7 00681000   031:0022:1
  IF Y[I2] > 0 THEN                                                00682000   031:0026:3
  BEGIN                                                            00683000   031:0027:3
   RATIO:=B[I2,-1]/Y[I2];                                        8 00684000   031:0028:0
   IF RATIO < MINRATIO THEN                                        00685000   031:002A:2
   BEGIN                                                           00686000   031:002A:5
    MINRATIO:=RATIO;                                             9 00687000   031:002B:2
    LEAVEROW:=I2;                                                  00688000   031:002C:1
   END;                                                            00689000   031:002D:1
  END;                                                           9 00690000   031:002D:1
  END;                                                           8 00691000   031:002D:1
  IF LEAVEROW = 0 AND MINRATIO ≠ 10**10 THEN                     7 00692000   031:002D:4
  BEGIN                                                            00693000   031:0030:2
   WRITE(LINE,<X5,"THE ENTERING COLUMN IS ",I3>,ENTERCOL);       7 00694000   031:0030:5
   WRITE(LINE,<X5,"@@@NO COLUMN IN THE TABLEAU IS ELIGIBLE TO",    00695000   031:0038:2
         " LEAVE THE BASIS AFTER ",I2," ITERATIONS THUS ",         00696000   031:003A:1
         "SUBSYSTEM #",I2," HAS AN UNBCLNDED SOLUTIONS ">,         00697000   031:003A:1
                                                                       DATA IS 0017 LONG
         ITERATION,I);                                             00698000   031:003B:0
   PROGRAMABORT;                                                   00699000   031:0040:2
  END;                                                             00700000   031:0041:3
 END OF OPTIMAL;                                                 7 00701000   031:0041:3
                                                                     B.0006(C31) IS 0047 LONG
END OF ZJMINUSCJ;                                                6 00702000   030:0068:3
                                                                   ZJMINUSCJ(C30) IS 0071 LONG
```

```
PROCEDURE UPDATE(B,BINDEX,BSIZE,ENTERCOL,LEAVEROW,Y);           5   02C:00C0:5
VALUE BSIZE,                                                        02C:00C0:5
      ENTERCOL,                                                     02C:00C0:5
      LEAVEROW;                                                     02C:00C0:5
INTEGER BSIZE,                                                      02C:00C0:5
        ENTERCOL,                                                   02C:00C0:5
        LEAVEROW;                                                   02C:00C0:5
INTEGER ARRAY BINDEX[0];                                            02C:00C0:5
REAL ARRAY B[0,-1],                                                 02C:00C0:5
           Y[0];                                                    02C:00C0:5
BEGIN                                                               02C:00C0:5
 REAL ARRAY FY[0:BSIZE],                                            02C:00C0:5
                                                                UPDATE IS SEGMENT 00033
             TEMP[-1:BSIZE];                                    5   033:0003:1
 INTEGER I3,                                                        033:0006:2
         J3,                                                        033:0006:2
         OLDINDEX;                                                  033:0006:2
% COMPUTE VECTOR NETA.                                              033:0006:2
 FOR I3:=0 STEP 1 UNTIL BSIZE DO                                    033:0006:2
 FY[I3]:=-Y[I3]/Y[LEAVEROW];                                        033:000A:5
 FY[LEAVEROW]:=1/Y[LEAVEROW]-1;                                     033:000E:0
 FOR I3:=-1 STEP 1 UNTIL BSIZE DO                                   033:0010:2
 BEGIN                                                              033:0015:0
  TEMP[I3]:=B[LEAVEROW,I3];                                     6   033:0015:0
  FOR J3:=0 STEP 1 UNTIL BSIZE DO                                   033:0017:4
  B[J3,I3]:=* + TEMP[I3] * FY[J3];                                  033:001C:1
 END;                                                               033:0020:5
% RESET VARIABLE X AS IT IS NO MORE IN THE BASIS.               6   033:0021:2
 X[OLDINDEX:=BINDEX[LEAVEROW]]:=0;                                  033:0021:2
 VARSTATUS[OLDINDEX]:=*-1;                                          033:0023:5
 BINDEX[LEAVEROW]:=ENTERCOL;                                        033:0026:0
% SET VALUE TO VARIABLE X IN THE BASIS.                             033:0027:3
 X[ENTERCOL]:=B[LEAVEROW,-1];                                       033:0027:3
 VARSTATUS[ENTERCOL]:=*+1;                                          033:0029:5
 IF VARSTATUS[ENTERCOL]=8 THEN                                      033:002C:0
 BEGIN                                                              033:002D:3
  WRITE(LINE,<X5,"@@@ARTIFICAL VARIABLE BEING PIVOT INTO THE ", 6   033:002E:0
        "BASIS THUS THE SUBSYSTEM #",I2," HAS AN INFEASIBLE ",      033:002F:5
        "SOLUTION">,I);                                             033:002F:5
                                                                    DATA IS 001F LONG
  PROGRAMABORT;                                                     033:0035:2
 END;                                                               033:0036:3
END OF UPDATE;                                                  6   033:0036:3
                                                                UPDATE(033) IS 0030 LONG
```

```
PROCEDURE FORMENTERCOL(CHECKOPTIMAL,I,MSIZE,MY);                    5   02C:00C0:5
VALUE I,                                                                02C:00C0:5
        MSIZE;                                                          02C:00C0:5
INTEGER I,                                                              02C:00C0:5
          MSIZE;                                                        02C:00C0:5
BOOLEAN CHECKOPTIMAL;                                                   02C:00C0:5
REAL ARRAY MY[1,0];                                                     02C:00C0:5
BEGIN                                                                   02C:00C0:5
 REAL ARRAY TEMP[2:MSIZE];                                              02C:00C0:5
                                                  FORMENTERCOL IS SEGMENT 00035
 INTEGER CONVEXROW,                                                 5   035:0003:1
           I5,                                                          035:0003:1
           J5;                                                          035:0003:1
 CONVEXROW:=NUMROWS[0] + I + 1;                                         035:0003:1
 FOR I5:=0 STEP 1 UNTIL MSIZE DO MY[I,I5]:= 0;                          035:0005:1
 FOR I5:=1 STEP 1 UNTIL NUMCOLS[I] DO                                   035:000C:1
 BEGIN                                                                  035:0011:0
  REAL WOA,                                                         6   035:0011:0
                                                        B.0007 IS SEGMENT 00036
        W1A;                                                            036:0000:1
  FOR J5:=1 STEP 1 UNTIL NUMROWS[0] DO                                  036:0000:1
  BEGIN                                                                 036:0004:5
   WOA:= * + M[0,J5+1] * C[I,J5,I5];%[M] * [A]                      7   036:0004:5
   W1A:=* + M[1,J5+1] * C[I,J5,I5];                                     036:0009:5
  END;                                                                  036:000E:5
  MY[I,0]:= * + (WOA + OBJ[I,I5]) * XVALUE[I,I5];                   7   036:000F:2
  MY[I,1]:= * + W1A * XVALUE[I,I5];                                     036:0015:3
  IF NOT CHECKOPTIMAL AND I5<=NUMROWS[0] THEN                           036:0019:5
  FOR J5:=1 STEP 1 UNTIL NUMCOLS[I] DO                                  036:001B:3
  MY[I,I5 + 1]:= * + C[I,I5,J5] * XVALUE[I,J5];                         036:002C:5
 END OF NUMCOEF;                                                        036:0026:1
                                                      B.0007(036) IS 002A LONG
 FOR I5:=0,1 DO MY[I,I5]:= * + M[I5,CONVEXROW];                     6   035:0012:1
 IF NOT CHECKOPTIMAL THEN MY[I,CONVEXROW]:=1;                           035:001C:3
 FOR I5:=2 STEP 1 UNTIL MSIZE DO                                        035:001F:2
 FOR J5:=2 STEP 1 UNTIL MSIZE DO                                        035:0024:0
 TEMP[I5]:= * + M[I5,J5] * MY[I,J5];                                    035:0028:4
 FOR I5:=2 STEP 1 UNTIL MSIZE DO MY[I,I5]:= TEMP[I5];                   035:002E:3
END OF FORMENTERCOL;                                                    035:0036:4
                                                 FORMENTERCOL(035) IS 003E LONG

PROCEDURE PIVOTENTERCOL(ARRAYSIZE,BIGX,I,MIT,MPHASE,MSIZE,          5   02C:00C0:5
                        XVALUE);                                        02C:00C0:5
VALUE ARRAYSIZE,                                                        02C:00C0:5
        I,                                                              02C:00C0:5
        MIT,                                                            02C:00C0:5
        MSIZE;                                                          02C:00C0:5
INTEGER ARRAYSIZE,                                                      02C:00C0:5
          I,                                                            02C:00C0:5
          MIT,                                                          02C:00C0:5
          MPHASE,                                                       02C:00C0:5
          MSIZE;                                                        02C:00C0:5
REAL ARRAY BIGX[2,1],                                                   02C:00C0:5
             XVALUE[1,1];                                               02C:00C0:5
BEGIN                                                                   02C:00C0:5
 REAL ARRAY FY[0:MSIZE],                                                02C:00C0:5
                                                PIVOTENTERCOL IS SEGMENT 00037
             TEMP[-1:MSIZE];                                        5   037:0003:1
 INTEGER I6,                                                            037:0006:2
           J6,                                                          037:0006:2
           K6,                                                          037:0006:2
           ROWLEAVE;                                                    037:0006:2
 REAL MINRATIO,                                                         037:0006:2
        RATIO;                                                          037:0006:2
% COMPUTE RATIO.                                                        037:0006:2
 MINRATIO:= 10**10;                                                     037:0006:2
 ROWLEAVE:=0;                                                           037:0008:3
 FOR I6:=2 STEP 1 UNTIL MSIZE DO                                        037:0009:1
 BEGIN                                                                  037:000D:5
 IF M[I6,-1] < RHSTOL THEN M[I6,-1]:= RHSTOL * I6;                  6   037:000D:5
 IF MY[I,I6] > 0 THEN                                                   037:0012:1
 BEGIN                                                                  037:0013:5
```

```
            RATIO:=M[I6,-1]/MY[I,I6];                                       7   00810000  037:0014:2
            IF RATIO < MINRATIO THEN                                            00811000  037:0017:2
            BEGIN                                                               00812000  037:0017:5
             MINRATIO:=RATIO;                                               8   00813000  037:0018:2
             ROWLEAVE:=I6;                                                      00814000  037:0019:1
            END;                                                                00815000  037:001A:1
           END;                                                             8   00816000  037:001A:1
           END;                                                             7   00817000  037:001A:1
           IF ROWLEAVE = 0 AND MINRATIO = 10**10 THEN                       6   00818000  037:001A:4
           BEGIN                                                                00819000  037:001D:2
            WRITE(LINE,<X5,"@@@NO COLUMN IN THE TABLEAU IS ELIGIBLE TO ",   6   00820000  037:001D:5
                  "LEAVE THE MASTER BASIS IN ITERATION #",I3," DURING ",        00821000  037:001F:4
                  "THE PIVOTING OF ENTERING COLUMN",/X8,"FROM SUBSYSTEM ",      00822000  037:001F:4
                  "#",I3," THUS THE PROBLEM HAS AN UNBOUNDED SOLUTION">,        00823000  037:001F:4
                                                                                    DATA IS 001B LONG
                 MIT,I);                                                        00824000  037:0020:3
            PROGRAMABORT;                                                       00825000  037:0026:2
           END;                                                                 00826000  037:0027:3
         % COMPUTE VECTOR NETA.                                             6   00827000  037:0027:3
           FOR J6:=0 STEP 1 UNTIL MSIZE DO                                      00828000  037:0027:3
           FY[J6]:= -MY[I,J6] / MY[I,ROWLEAVE];                                 00829000  037:002C:0
           FY[ROWLEAVE]:=1/MY[I,ROWLEAVE]-1;                                    00830000  037:0030:3
           FOR J6:=-1 STEP 1 UNTIL MSIZE DO                                     00831000  037:0033:3
           BEGIN                                                                00832000  037:0038:1
            TEMP[J6]:=M[ROWLEAVE,J6];                                       6   00833000  037:0038:1
            FOR K6:=0 STEP 1 UNTIL MSIZE DO                                     00834000  037:003A:5
            M[K6,J6]:= * + TEMP[J6] * FY[K6];                                   00835000  037:003F:2
           END;                                                                 00836000  037:0044:0
           IF MP[6] THEN                                                    6   00837000  037:0044:3
           WRITE(LINE,<"THE INCOMING CANDIDATE COLUMN IS GENERATED FROM",       00838000  037:0045:1
                 " SUBSYSTEM #",I3>,I);                                         00839000  037:0047:3
                                                                                    DATA IS 0031 LONG
           IF MP[1] THEN                                                        00840000  037:004C:2
           BEGIN                                                                00841000  037:004C:5
            WRITE(LINE,<X5,"@@@MASTER PROBLEM HAS NOT REACHED OPTIMAL ",    6   00842000  037:004D:2
                   "AFTER ITERATION #",I3,/X5,"AND THE LEAVING WEIGHT IS "      00843000  037:004F:1
                   ,"W[",I5,"]",/X9,"THE ENTERING WEIGHT IS W[",I5,"]">,        00844000  037:004F:1
                                                                                    DATA IS 000F LONG
                  MIT,WEIGHT[ROWLEAVE],-(MIT*100+I));                           00845000  037:0050:0
            WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",       00846000  037:0059:2
                  F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  ="       00847000  037:005B:1
                  ,F13.6>,M[0,-1],M[1,-1]);                                     00848000  037:005B:1
                                                                                    DATA IS 0028 LONG
            WRITE(LINE,<"THE PIVOT ELEMENT =",F20.6>,MY[I,ROWLEAVE]);           00849000  037:0064:2
           END;                                                                 00850000  037:006D:2
           WEIGHT[ROWLEAVE]:=-(MIT * 100 + I);                              6   00851000  037:006D:2
           RESIZE(BIGX[ROWLEAVE, * ], ARRAYSIZE, RETAIN);                       00852000  037:0070:3
           RESIZE(XVALUE[I,*], ARRAYSIZE, RETAIN);                              00853000  037:0073:2
           FOR K6:=1 STEP 1 UNTIL ARRAYSIZE DO                                  00854000  037:0076:0
           BIGX[ROWLEAVE,K6]:=XVALUE[I,K6];                                     00855000  037:007A:3
           XSIZE[I]:=ARRAYSIZE;                                                 00856000  037:007F:0
     $ SET OMIT = NOPRINT                                                       00857000    OMIT
           WRITE(LINE,<"BIGX",I5(/,10(F13.6))>,BIGX[ROWLEAVE,*]);               00858000    OMIT
     $ POP OMIT                                                                 00859000  037:0080:5
          END OF PIVOTENTERCOL;                                                 00860000  037:0080:5
                                                                      PIVOTENTERCOL(037) IS 0090 LONG
```

```
PROCEDURE MATRIXREINVERSION(BASIS,I,INDEX,PRODUCTPIVOT,SIZE);      5   02C:00C0:5
VALUE I,                                                                02C:00C0:5
          SIZE;                                                         02C:00C0:5
INTEGER I,                                                              02C:00C0:5
          SIZE;                                                         02C:00C0:5
REAL PRODUCTPIVOT;                                                      02C:00C0:5
INTEGER ARRAY INDEX[0];                                                 02C:00C0:5
REAL ARRAY BASIS[0,-1];                                                 02C:00C0:5
BEGIN                                                                   02C:00C0:5
 INTEGER ARRAY DUMMYOBJ[0:2*SIZE+1]; % DUMMY OBJECTIVE.                 02C:00C0:5
                                       MATRIXREINVERSION IS SEGMENT 0003C
 REAL ARRAY NEWB[0:SIZE,-2:2*SIZE+1];                              5   03C:0004:0
 BOOLEAN FOUND;                                                         03C:0009:0
 INTEGER ECOL, % THE INDEX FOR ENTERING COLUMN.                         03C:0009:0
         I8,   % LOOP COUNTER.                                          03C:0009:0
         J8,                                                            03C:0009:0
         K8,                                                            03C:0009:0
         L8,                                                            03C:0009:0
         LROW; % THE INDEX FOR LEAVING ROW.                             03C:0009:0
 REAL MINRATIO,                                                         03C:0009:0
      MINZJCJ,                                                          03C:0009:0
      RATIO,                                                            03C:0009:0
      ZJCJ;                                                             03C:0009:0
 DEFINE UPTO=STEP 1 UNTIL #;                                            03C:0009:0
 IF S[I,7] THEN                                                         03C:0009:0
 FOR I8:= 0 UPTO SIZE DO                                                03C:000A:2
 WRITE(LINE,<4(/,10(F13.6))>,FOR L8:=-1 UPTO SIZE DO                    03C:000F:2
          BASIS[I8,L8]);                                                03C:0017:2
 NEWB[0,0]:=NEWB[1,1]:=1;                                               03C:001D:5
 FOR I8:=2 UPTO SIZE DO                                                 03C:0020:4
 BEGIN                                                                  03C:0025:2
  J8:=INDEX[I8];                                                   6   03C:0025:2
  NEWB[0,I8]:=OBJCOEFF[L,I,J8];                                         03C:0026:4
  IF VARSTATUS[J8] = 8 THEN NEWB[1,I8]:= 1;                             03C:002A:3
  FOR K8:=1 UPTO NUMROWS[I] DO                                          03C:002E:3
  NEWB[K8+1,I8]:=A[I,K8,J8];                                            03C:0033:2
  NEWB[I8,-1]:= RHS[I,I8-1];                                            03C:003B:3
  NEWB[I8,-2]:=-1;                                                      03C:003B:3
 END;                                                                   03C:003D:2
%                                                                  6   03C:003D:5
% ATTACHED ARTIFICIAL VARIABLES TO ORIGINAL MATRIX.                     03C:003D:5
 J8:= SIZE + 1;                                                         03C:003D:5
 K8:= 2*SIZE + 1;                                                       03C:003F:1
 FOR I8:=J8 UPTO K8 DO                                                  03C:0041:0
 BEGIN                                                                  03C:0045:0
  NEWB[I8-J8,I8]:=1;                                               6   03C:0045:0
  DUMMYOBJ[I8]:=-1;                                                     03C:0047:4
 END;                                                                   03C:0049:0
% REINITIALIZE THE PRODUCT OF PIVOT ELEMENTS.                      6   03C:0049:3
 PRODUCTPIVOT:=1;                                                       03C:0049:3
% COMPUTE ZJ-CJ FOR ALL COLUMNS.                                        03C:004A:2
 FOUND:=FALSE;                                                          03C:004A:2
 WHILE NOT FOUND DO                                                     03C:004B:0
 BEGIN                                                                  03C:004B:3
  FOUND:=TRUE;                                                     6   03C:004B:5
  MINZJCJ:=-ZJCJTOL;                                                    03C:004C:3
  FOR I8:=0 UPTO K8 DO                                                  03C:004D:3
  BEGIN                                                                 03C:0052:0
   ZJCJ:=0;                                                        7   03C:0052:0
   FOR L8:=2 UPTO SIZE DO                                               03C:0052:4
   ZJCJ:= * + NEWB[L8,-2] * NEWB[L8,I8];                                03C:0057:2
   ZJCJ:=*-DUMMYOBJ[I8];                                                03C:005B:3
   IF ZJCJ < MINZJCJ THEN                                               03C:005D:1
   BEGIN                                                                03C:005D:4
    MINZJCJ:=ZJCJ;                                                 8   03C:005E:1
    ECOL:=I8;                                                           03C:005F:0
    FOUND:=FALSE;                                                       03C:0060:0
   END;                                                                 03C:0060:4
  END;                                                             8   03C:0060:4
  MINRATIO:= 10**10;                                               7   03C:0061:1
  IF NOT FOUND THEN                                                     03C:0063:3
  BEGIN                                                                 03C:0063:5
 % COMPUTE RATIO TEST.                                             7   03C:0064:2
   FOR I8:=2 UPTO SIZE DO                                               03C:0064:2
   BEGIN                                                                03C:0069:0
   IF NEWB[I8,-1] < RHSTOL THEN NEWB[I8,-1]:= RHSTOL * I8;         8   03C:0069:0
   IF NEWB[I8,ECOL] > 0 THEN                                            03C:006D:0
```

```
        BEGIN                                                                00937000   03C:006E:5
         RATIO:= NEWB[I8,-1] / NEWB[I8,ECOL];                           9    00938000   03C:006F:2
         IF RATIO < MINRATIO THEN                                            00939000   03C:0072:2
         BEGIN                                                               00940000   03C:0072:5
          MINRATIO:=RATIO;                                              10   00941000   03C:0073:2
          LROW:=I8;                                                          00942000   03C:0074:1
         END;                                                                00943000   03C:0075:1
        END;                                                            10   00944000   03C:0075:1
        END;                                                             9   00945000   03C:0075:1
        PRODUCTPIVOT:=PRODUCTPIVOT*NEWB[LROW,ECOL];                      8   00946000   03C:0075:4
      % UPDATE MATRIX IN ORDER TO OBTAIN THE NEW BASIS INVERSE.              00947000   03C:0078:2
        FOR I8:=0 UPTO LROW-1, LROW+1 UPTO SIZE DO                           00948000   03C:0078:2
        FOR L8:=-1, 2 UPTO ECOL-1, ECOL+1 UPTO K8 DO                         00949000   03C:0084:3
        NEWB[I8,L8]:= * -NEWB[LROW,L8] * NEWB[I8,ECOL]                       00950000   03C:0093:3
                        /NEWB[LROW,ECOL];                                    00951000   03C:0098:5
        FOR I8:=0 UPTO LROW-1,LROW+1 UPTO SIZE DO NEWB[I8,ECOL]:=0;          00952000   03C:00A0:3
        FOR L8:=-1 UPTO ECOL-1, ECOL+1 UPTO K8 DO                            00953000   03C:00B1:3
        NEWB[LROW,L8]:= */NEWB[LROW,ECOL];                                   00954000   03C:00BE:0
        NEWB[LROW,ECOL]:=1;                                                  00955000   03C:00C5:3
        NEWB[LROW,-2]:=DUMMYOBJ[ECOL];                                       00956000   03C:00C7:4
       END OF FI;                                                            00957000   03C:00C9:5
      END OF EILHW;                                                      7   00958000   03C:00C9:5
      IF S[I,6] THEN                                                     6   00959000   03C:00CA:2
      WRITE(LINE,<"MATRIX REINVERSION TAKES PLACE AND THE NEW ",             00960000   03C:00CB:4
            "PRODUCT OF PIVOT ELEMENTS IS:",F20.6>,PRODUCTPIVOT);            00961000   03C:00CE:0
                                                                                 DATA IS 0025 LONG
      FOR I8:=0 UPTO SIZE DO                                                 00962000   03C:00D3:2
      BEGIN                                                                  00963000   03C:00D7:5
       BASIS[I8,-1]:=NEWB[I8,-1];                                        6   00964000   03C:00D7:5
       FOR L8:= 0 UPTO SIZE DO BASIS[I8,L8]:= NEWB[I8,J8 + L8];              00965000   03C:00DA:1
      END;                                                                   00966000   03C:00E3:0
      IF S[I,7] THEN                                                     6   00967000   03C:00E3:3
      FOR I8:=0 UPTO SIZE DO                                                 00968000   03C:00E4:5
      WRITE(LINE,<4(/,10(F13.6))>,FOR L8:=-1 UPTO SIZE DO                    00969000   03C:00E9:5
            BASIS[I8,L8]);                                                   00970000   03C:00F1:5
     END OF MATRIXREINVERSION;                                               00971000   03C:00F8:5
                                                          MATRIXREINVERSION(03C) IS 0104 LONG
```

```
% SET UP MASTER PROBLEM.                                                 5   02C:00C0:5
IF ADVANCESOLN = 0 THEN                                                      02C:00C0:5
BEGIN                                                                        02C:00C1:3
M[0,0]:=M[1,1]:=1;                                                       5   02C:00C2:0
K4:=1;                                                                       02C:00C4:4
FOR I4:=1 STEP 1 UNTIL NUMROWS[0] DO                                         02C:00C5:2
CASE STATUS[0,I4] + 2 OF                                                     02C:00CA:0
BEGIN                                                                        02C:00CB:4
 1: M[1,K4:= * + 1]:=-1;                                                 6   02C:00CB:4
    M[K4,-1]:=RHS[0,I4];                                                     02C:00D1:4
    M[K4,K4]:=1;                                                             02C:00D4:2
 2: M[1,K4:= * + 1]:=-1;                                                     02C:00D6:2
    M[K4,-1]:=RHS[0,I4];                                                     02C:00D9:5
    M[K4,K4]:=1;                                                             02C:00DC:3
 3: M[K4:= * + 1,-1]:=RHS[0,I4];                                             02C:00DE:3
    M[K4,K4]:=1;                                                             02C:00E2:4
END;                                                                         02C:00E4:4
FOR I4:=1 STEP 1 UNTIL NUMBLOCKS DO                                      6   02C:00E7:3
BEGIN                                                                        02C:00EC:0
 M[K4:= * + 1,-1]:=1;                                                    6   02C:00EC:0
 M[1,K4]:= -1;                                                               02C:00EE:4
 M[K4,K4]:=1;                                                                02C:00F0:4
END;                                                                         02C:00F2:4
FOR I4:=0,1 DO                                                           6   02C:00F3:1
FOR L4:=0 STEP 1 UNTIL MSIZE DO                                              02C:00F7:1
TEMPX[I4]:= * + M[I4,L4] * M[L4,-1];                                         02C:00F8:4
FOR I4:=0,1 DO M[I4,-1]:=TEMPX[I4];                                          02C:0103:3
IF MP[0] THEN                                                                02C:010C:3
WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",                02C:010D:0
      F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =",               02C:010F:2
      F13.6>,M[0,-1],M[1,-1]);                                               02C:010F:2
                                                               DATA IS 0018 LONG
END                                                                          02C:0118:2
ELSE BEGIN                                                               5   02C:0118:2
     % READ IN NUMBER OF VARIABLES PER BLOCK.                            5   02C:0118:5
       READ(ADVANCEFILE,*,NUMVARS[*]);                                       02C:0118:5
       WRITE(LINE,/,NUMVARS[*]);                                             02C:0123:2
       FOR I4:=1 UPTO NUMBLOCKS DO                                           02C:012C:2
       BEGIN                                                                 02C:0130:5
        J4:= NUMROWS[I4] + 1;                                            6   02C:0130:5
      % READ IN THE OPTIMAL TABLEAU FOR EACH SUBSYSTEM.                      02C:0132:3
        FOR K4:=0 UPTO J4 DO                                                 02C:0132:3
        BEGIN                                                                02C:0137:0
         READ(ADVANCEFILE,*,FOR L4:=-1 UPTO J4 DO OTS[I4,K4,L4]);        7   02C:0137:0
         WRITE(LINE,/,FOR L4:=-1 UPTO J4 DO OTS[I4,K4,L4]);                  02C:0148:2
        END;                                                                 02C:0157:2
      % READ IN THE INDICES OF VARIABLES IN THE BASIS OF EACH            7   02C:0157:5
      % SUBSYSTEM.                                                           02C:0157:5
        READ(ADVANCEFILE,*,FOR L4:=0 UPTO J4 DO OTINDEX[I4,L4]);             02C:0157:5
        WRITE(LINE,/,FOR L4:=0 UPTO J4 DO OTINDEX[I4,L4]);                   02C:0168:2
      % READ IN THE OPTIMAL SOLUTION FOR THE SUBSYSTEM.                      02C:0176:2
        READ(ADVANCEFILE,*,FOR L4:=1 UPTO NUMVARS[I4] DO                     02C:0176:2
             XVALUE[I4,L4]);                                                 02C:017E:2
        WRITE(LINE,/,FOR L4:=1 UPTO NUMVARS[I4] DO XVALUE[I4,L4]);           02C:0187:2
       END;                                                                  02C:0196:2
     % READ IN THE BASIS FOR MASTER PROBLEM.                             6   02C:0196:5
       FOR I4:=0 UPTO MSIZE DO                                               02C:0196:5
       BEGIN                                                                 02C:019B:2
        READ(ADVANCEFILE,*,FOR L4:=-1 UPTO MSIZE DO M[I4,L4]);           6   02C:019B:2
        WRITE(LINE,/,FOR L4:=-1 UPTO MSIZE DO M[I4,L4]);                     02C:01A8:2
       END;                                                                  02C:01B9:2
       READ(ADVANCEFILE,*,WEIGHT[*]);                                    6   02C:01B9:5
       WRITE(LINE,/,WEIGHT[*]);                                              02C:01C4:2
       FOR I4:=2 UPTO MSIZE DO                                               02C:01CD:2
       BEGIN                                                                 02C:01D2:0
        J4:= -(WEIGHT[I4] MOD 100);                                      6   02C:01D2:0
        IF J4 > 0  AND  J4 <= NUMBLOCKS THEN                                 02C:01D4:3
        BEGIN                                                                02C:01D5:5
        READ(ADVANCEFILE,*,FOR L4:=1 UPTO NUMVARS[J4] DO                 7   02C:01D6:2
             BIGX[I4,L4]);                                                   02C:01DE:2
        WRITE(LINE,/,FOR L4:=1 UPTO NUMVARS[J4] DO BIGX[I4,L4]);             02C:01E7:2
        END;                                                                 02C:01F6:2
       END;                                                              7   02C:01F6:2
      END;                                                               6   02C:01F6:5
IF M[1,-1]=0 THEN MPHASE:=0 ELSE MPHASE:=1;                              5   02C:01F6:5
MASTEROPTIMAL:=FALSE;                                                        02C:01FA:2
WHILE NOT MASTEROPTIMAL AND MIT <= QUITERATION DO                            02C:01FB:0
```

```
        BEGIN                                                                          02C:01FC:3
         IF MP[7] THEN                                                           5     02C:01FD:0
         BEGIN                                                                         02C:01FD:4
          WRITE(LINE,<"THE DUAL PRICES FOR THE COUPLING ROWS ARE:">);            6     02C:01FE:1
                                                                                  DATA IS 0017 LONG
           FOR K:=1 STEP 6 UNTIL NUMROWS[0] DO                                         02C:0203:2
           WRITE(LINE,<6("CUP",I3,X1,F13.6,X2)>,FOR N:=K STEP 1 WHILE                  02C:0208:1
                 (N <= K+5  AND  N <= NUMROWS[0]) DO [N,M[1,N+1]]);                    02C:020D:3
           WRITE(LINE,<"THE DUAL PRICES FOR THE CONVEXITY ROWS ARE:">);                02C:0219:5
                                                                                  DATA IS 0013 LONG
           FOR K:=1 STEP 6 UNTIL NUMBLOCKS DO                                          02C:021E:2
           WRITE(LINE,<6("CNX",I3,X1,F13.6,X2)>,FOR N:=K STEP 1 WHILE                  02C:0223:0
                 (N <= K+5  AND  N <= NUMBLOCKS) DO [N,M[1,NUMROWS[0]+N+1]]            02C:0228:3
                 );                                                                    02C:0231:3
         END;                                                                          02C:0234:5
        FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO                                       6     02C:0234:5
        BEGIN                                                                          02C:0239:2
         INTEGER NUMCOEFF;                                                       6     02C:0239:2
                                                                                B.0008 IS SEGMENT 00043
         ARRAYSIZE:=NUMCOLS[I]+2*NUMROWS[I];                                           043:0000:1
         BSIZE:=NUMROWS[I]+1;                                                          043:0002:5
         NUMCOEFF:=NUMCOLS[I];                                                         043:0004:3
         BEGIN                                                                         043:0005:5
          INTEGER ARRAY BINDEX[0:BSIZE];                                         7     043:0005:5
                                                                                B.0009 IS SEGMENT 00044
          REAL ARRAY B[0:BSIZE,-1:BSIZE],                                              044:0003:1
                     Y[0:BSIZE];                                                       044:0007:2
          INTEGER ITERATION;         % COMPUTE FOR NUMBER OF ITERATION.                044:000A:2
          PRODUCTPIVOT :=1;                                                            044:000A:2
          IF MIT = 0 AND ADVANCESOLN = 0 THEN                                          044:000B:0
          BEGIN                                                                        044:000C:3
          FOR J:=1 STEP 1 UNTIL NUMCOLS[I] DO                                    8     044:000D:0
          BEGIN                                                                        044:0011:5
           X[J]:=0;                                                              9     044:0011:5
           VARSTATUS[J]:=1;        % REAL VARIABLE                                     044:0013:2
          END;                                                                         044:0014:5
          ADDSLASURART(I);                                                       9     044:0015:2
          NUMVARS[I]:= ARRAYSIZE;                                                      044:0016:3
          WRITE(LINE,<X5,"@@@SLACK, SURPLUS AND ARTIFICIAL VARIABLES H",               044:0018:2
                "AVE BEEN ADDED TO VARIOUS CONSTRAINS FOR SUBSYSTEM #",I3              044:001A:1
                >,I);                                                                  044:001A:1
                                                                                  DATA IS 0014 LONG
          WRITE(LINE,<"TOTAL NUMBER OF VARIABLES IN THIS SUBSYSTEM IS =",              044:001F:2
                      I3>,ARRAYSIZE);                                                  044:0021:1
                                                                                  DATA IS 0017 LONG
$ SET OMIT = NOPRINT                                                                     OMIT
          WRITE(LINE,<"THE ORIGINAL MATRIX 'A' SET UP FOR THIS SUBSYSTEM"                OMIT
                ," IS :">);                                                              OMIT
          FOR N:=1 STEP 1 UNTIL NUMROWS[I] DO                                            OMIT
          WRITE(LINE,<15(/,10(F13.6))>,A[I,N,*]);                                        OMIT
$ POP OMIT                                                                     044:0026:2
          SETUPBASIS(ARRAYSIZE,B,BINDEX,I);                                            044:0026:2
$ SET OMIT = NOPRINT                                                                     OMIT
          WRITE(LINE,<X5,"@@@THE INVERSE OF INITIAL BASIS B HAS BEEN ",                  OMIT
                "SET UP FOR SUBSYSTEM #",I3," AND IS OF SIZE",I3," BY",                  OMIT
                I3>,I,BSIZE+1,BSIZE+1);                                                  OMIT
          FOR N:=0 STEP 1 UNTIL BSIZE DO                                                 OMIT
          WRITE(LINE,<4(/,10(F13.6))>,B[N,*]);                                           OMIT
$ POP OMIT                                                                     044:0029:0
          IF S[I,0] THEN                                                               044:0029:0
          BEGIN                                                                        044:002A:1
          WRITE(LINE,<"THE INDICES OF VARIABLES IN THE BASIS ARE :">);           9     044:002A:4
                                                                                  DATA IS 000C LONG
          WRITE(LINE,<33I4>,BINDEX[*]);                                                044:002F:2
          FOR N:=2 STEP 6 UNTIL BSIZE DO                                               044:0038:2
          WRITE(LINE,<6("X[",I3,"]",X1,F13.6,X2)>,FOR J:=N STEP 1 WHILE                044:003D:1
                (J <= N+5 AND J <= BSIZE) DO [BINDEX[J],X[BINDEX[J]]]);                044:0042:3
          WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",                044:004E:5
                F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =",               044:0050:4
                F13.6>,B[0,-1],B[1,-1]);                                               044:0050:4
                                                                                  DATA IS 0019 LONG
          END;                                                                         044:0059:2
          PHASE:=1;                                                              9     044:0059:2
          END                                                                          044:005A:0
          ELSE FOR J:=0 STEP 1 UNTIL BSIZE DO                                    8     044:005A:0
                BEGIN                                                                  044:005F:0
                 FOR K:=-1 STEP 1 UNTIL BSIZE DO                                 8     044:005F:0
```

```
        B[J,K]:=OTS[I,J,K];
        BINDEX[J]:=OTINDEX[I,J];
       END;
%
%
%
  ARRAYSIZE:= NUMVARS[I];
  RESIZE(OBJ[I,*],ARRAYSIZE,RETAIN);
  FOR J:=1 STEP 1 UNTIL NUMCOEFF DO
  BEGIN
   OBJ[I,J]:=OBJCOEFF[L,I,J];
   IF MPHASE=1 THEN OBJCOEFF[L,I,J]:=0;
   FOR K:=1 STEP 1 UNTIL NUMROWS[0] DO
   OBJCOEFF[L,I,J]:= * + M[MPHASE,K+1] * C[I,K,J];
  END;
  FOR J:=-1, 2 STEP 1 UNTIL BSIZE DO
  BEGIN
   B[0,J]:=0;
   FOR K:=2 STEP 1 UNTIL BSIZE DO
   B[0,J]:= * + ( - OBJCOEFF[L,I,BINDEX[K]]) * B[K,J];
  END;
  ITERATION:=ENTERCOL:=LEAVEROW:=0;
  ZJMINUSCJ(ARRAYSIZE,B,BINDEX,BSIZE,I,ITERATION,PHASE,Y);
  WHILE PHASE=1 OR NOT OPTIMAL DO
  BEGIN
   IF NOT OPTIMAL THEN
   BEGIN
    ITERATION:=*+1;
    IF S[I,1] THEN
    WRITE(LINE,<X5,"@@@SUBSYSTEM #",I2," HAS NOT REACHED OPTIMAL"
         ," AFTER ITERATION #",I2,/X5,"AND THE LEAVING VECTOR IS"
         ," X[",I3,"]",/X9,"THE ENTERING VECTOR IS X[",I3,"]">,I,

          ITERATION,BINDEX[LEAVEROW],ENTERCOL);
    UPDATE(B,BINDEX,BSIZE,ENTERCOL,LEAVEROW,Y);
    PRODUCTPIVOT:= PRODUCTPIVOT * Y[LEAVEROW];
    IF S[I,1] THEN
    WRITE(LINE,<X5,"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II"
          ," =",F20.6,/X5,"THE VALUE OF OBJECTIVE FUNCTION FOR ",
          "PHASE I  =",F20.6,/X5,"THE PRODUCT OF PIVOT ELEMENTS "
          ,"=",X15,F20.6,/X5,"AND THE SUBSYSTEM IS IN PHASE =",I2
          >,B[0,-1],B[1,-1],PRODUCTPIVOT,PHASE);

    IF PRODUCTPIVOT <= MBXTOL OR ITERATION >= TOLITERATION THEN
    MATRIXREINVERSION(B,I,BINDEX,PRODUCTPIVOT,BSIZE);
   END;
   IF PHASE = 1 AND ABS(B[1,-1]) <= MBXTOL THEN
   BEGIN
    PHASE:=0;
    OPTIMAL:=FALSE;
    IF S[I,2] THEN
    BEGIN
     WRITE(LINE,<"THE INDICES OF VARIABLES IN THE BASIS ARE:">);

     WRITE(LINE,<33I4>,BINDEX[*]);
     FOR N:=2 STEP 6 UNTIL BSIZE DO
     WRITE(LINE,<6("X[",I3,"]",X1,F13.6,X2)>,FOR K:=N STEP 1
           WHILE (K <= N+5 AND K <= BSIZE) DO [BINDEX[K],
           X[BINDEX[K]]]);
     WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II ="
           ,F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I "
           ," =",F13.6>,B[0,-1],B[1,-1]);

    END;
    IF S[I,3] THEN
    BEGIN
     WRITE(LINE,<"THE BASIS INVERSE AT END OF PHASE I IS AS ",
           "FOLLOW:">);

     FOR N:=0 STEP 1 UNTIL BSIZE DO
     WRITE(LINE,<4(/,10(F13.6))>,B[N,*]);
    END;
   END;
   ZJMINUSCJ(ARRAYSIZE,B,BINDEX,BSIZE,I,ITERATION,PHASE,Y);
  END;
  IF S[I,4] THEN
  BEGIN
```

```
      01117000   044:0063:4
      01118000   044:0068:0
      01119000   044:006A:3
 8    01120000   044:006B:0
      01121000   044:006B:0
      01122000   044:006B:0
      01123000   044:006B:0
      01124000   044:006C:4
      01125000   044:006F:2
      01126000   044:0073:5
 8    01127000   044:0073:5
      01128000   044:0078:0
      01129000   044:007C:0
      01130000   044:0080:4
      01131000   044:0088:3
 8    01132000   044:0089:0
      01133000   044:0091:2
 8    01134000   044:0091:2
      01135000   044:0093:1
      01136000   044:0097:5
      01137000   044:009E:3
 8    01138000   044:00A1:3
      01139000   044:00A3:3
      01140000   044:00A8:1
      01141000   044:00A9:3
 8    01142000   044:00AA:0
      01143000   044:00AA:2
 9    01144000   044:00AA:5
      01145000   044:00AC:1
      01146000   044:00AD:2
      01147000   044:00AF:4
      01148000   044:00AF:4
              DATA IS 0017 LONG
      01149000   044:00B2:2
      01150000   044:00B9:2
      01151000   044:00BD:0
      01152000   044:00BE:4
      01153000   044:00BF:5
      01154000   044:00C2:1
      01155000   044:00C2:1
      01156000   044:00C2:1
      01157000   044:00C2:1
              DATA IS 002A LONG
      01158000   044:00CD:2
      01159000   044:00CF:1
      01160000   044:00D2:5
 9    01161000   044:00D2:5
      01162000   044:00D5:2
 9    01163000   044:00D5:5
      01164000   044:00D6:3
      01165000   044:00D7:1
      01166000   044:00D8:3
10    01167000   044:00D9:0
              DATA IS 0032 LONG
      01168000   044:00DD:2
      01169000   044:00E6:2
      01170000   044:00EB:1
      01171000   044:00EF:2
      01172000   044:00F6:3
      01173000   044:00FC:5
      01174000   044:00FE:4
      01175000   044:00FE:4
              DATA IS 0018 LONG
      01176000   044:0107:2
10    01177000   044:0107:2
      01178000   044:0108:4
10    01179000   044:0109:1
      01180000   044:010B:0
              DATA IS 0019 LONG
      01181000   044:010E:2
      01182000   044:0112:5
      01183000   044:011B:5
10    01184000   044:011B:5
 9    01185000   044:011B:5
      01186000   044:0120:3
 8    01187000   044:0121:0
      01188000   044:0122:2
```

```
            WRITE(LINE,<X5,"@@@OPTIMAL HAS REACHED FOR SUBSYSTEM #",I2,          8   01189000  044:0122:5
                 " AFTER ITERATION #",I2," AND THE SOLUTIONS ARE:">,I,               01190000  044:0124:4
                                                                                         DATA IS 0013 LONG
                 ITERATION + 1);                                                     01191000  044:0127:2
            WRITE(LINE,<"THE INDICES OF VARIABLES IN THE BASIS ARE:">);              01192000  044:012B:2
                                                                                         DATA IS 0016 LONG
            WRITE(LINE,<33I4>,BINDEX[*]);                                            01193000  044:0130:2
$ SET OMIT = READISK                                                                 01194000  044:0139:2
$ SET LIST = NOT READISK                                                             01195000  044:0139:2
           FOR J:=1 STEP 4 UNTIL NUMCOLS[I] DO                                       01196000  044:0139:2
           WRITE(LINE,  <4("X[",I3,"]",X4,C6,X4,F13.6)>,FOR K:=J STEP 1              01197000  044:013E:2
                 WHILE (K <= J+3 AND K <= NUMCOLS[I]) DO [K,VARNAME[I,K],            01198000  044:0142:3
                 X[K]]);                                                             01199000  044:014C:0
$ POP OMIT LIST                                                                      01200000  044:0151:5
$ SET OMIT = NOT READISK                                                             01201000     OMIT
$ POP OMIT LIST                                                                      01206000  044:0151:5
            WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",            01207000  044:0151:5
                 F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =",            01208000  044:0153:4
                 F13.6>,B[0,-1],B[1,-1]);                                            01209000  044:0153:4
                                                                                         DATA IS 0019 LONG
           END;                                                                      01210000  044:015C:2
           IF S[I,5] THEN                                                        8   01211000  044:015C:2
           BEGIN                                                                     01212000  044:015D:4
            WRITE(LINE,<"THE BASIS INVERSE AT END OF PHASE II IS AS ",           8   01213000  044:015E:1
                 "FOLLOW:">);                                                        01214000  044:0160:0
                                                                                         DATA IS 0017 LONG
            FOR N:=0 STEP 1 UNTIL BSIZE DO                                           01215000  044:0163:2
            WRITE(LINE,<4(/,10(F13.6))>,B[N,*]);                                     01216000  044:0167:5
           END;                                                                      01217000  044:0170:5
           FOR J:=0 STEP 1 UNTIL BSIZE DO                                        8   01218000  044:0170:5
           BEGIN                                                                     01219000  044:0175:2
            FOR K:=-1 STEP 1 UNTIL BSIZE DO                                      8   01220000  044:0175:2
            OTS[I,J,K]:=B[J,K];                                                      01221000  044:017A:0
            OTINDEX[I,J]:=BINDEX[J];                                                 01222000  044:017E:3
           END;                                                                      01223000  044:0181:0
           RESIZE(XVALUE[I,*],ARRAYSIZE,RETAIN);                                 8   01224000  044:0181:3
           FOR J:=1 STEP 1 UNTIL ARRAYSIZE DO                                        01225000  044:0184:1
           BEGIN                                                                     01226000  044:0188:4
            XVALUE[I,J]:= X[J];                                                  8   01227000  044:0188:4
            X[J]:= 0;                                                                01228000  044:018B:5
           END;                                                                      01229000  044:018D:2
         % TO COMPUTE RELATIVE COST FACTOR RCOSTF.                               8   01230000  044:018D:5
           RCOSTF[I]:= M[MPHASE, NUMROWS[0] + I + 1];                                01231000  044:018D:5
           FOR J:=1 STEP 1 UNTIL NUMCOEFF DO                                         01232000  044:0191:3
           BEGIN                                                                     01233000  044:0196:0
            RCOSTF[I]:= * + OBJCOEFF[L,I,J] * XVALUE[I,J];                       8   01234000  044:0196:0
            OBJCOEFF[L,I,J]:= OBJ[I,J];                                              01235000  044:019B:4
           END;                                                                      01236000  044:01A0:0
          END;                                                                   8   01237000  044:01A0:3
                                                                                   B.0009(044) IS 01C5 LONG
         END OF BLKS;                                                            7   01238000  043:0006:3
                                                                                   B.0008(043) IS 0009 LONG
         CHECKOPTIMAL:= FALSE;                                                   6   01239000  02C:023A:3
         MIT:= * + 1;                                                                01240000  02C:023B:1
         NOENTERCANDIDATE:= TRUE;                                                    01241000  02C:023C:3
         FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO                                          01242000  02C:023D:1
         IF RCOSTF[I] < -ZJCJTOL THEN                                                01243000  02C:0241:4
         BEGIN                                                                       01244000  02C:0243:2
          NOENTERCANDIDATE:= FALSE;                                              6   01245000  02C:0243:5
          K:= I;                                                                     01246000  02C:0244:3
          FOR J:=1 STEP 1 UNTIL NUMBLOCKS DO                                         01247000  02C:0245:3
          IF RCOSTF[J] < RCOSTF[I] THEN I:= J;                                       01248000  02C:024A:0
          FORMENTERCOL(CHECKOPTIMAL,I,MSIZE,MY);                                     01249000  02C:024E:1
          ARRAYSIZE:= NUMVARS[I];                                                    01250000  02C:0250:5
          IF MY[I,MPHASE] < 0 THEN                                                   01251000  02C:0252:3
          PIVOTENTERCOL(ARRAYSIZE,BIGX,I,MIT,MPHASE,MSIZE,XVALUE);                   01252000  02C:0254:1
          RCOSTF[I]:= 0;                                                             01253000  02C:0258:5
          I:= K - 1;                                                                 01254000  02C:025A:2
         END;                                                                        01255000  02C:025B:4
         IF NOENTERCANDIDATE THEN                                                6   01256000  02C:025C:1
         IF MPHASE = 0 THEN MASTEROPTIMAL := TRUE                                    01257000  02C:025C:3
         ELSE BEGIN                                                                  01258000  02C:025E:1
               WRITE(LINE,<X5,"NO INCOMING CANDIDATE COLUMN GENERATED ",         6   01259000  02C:025F:2
                     "FROM SUBSYSTEMS WHEN IN PHASE I THUS MASTER PROBLEM"           01260000  02C:0261:1
                     ," HAS AN INFEASIBLE SOLUTION">);                               01261000  02C:0261:1
                                                                                         DATA IS 0014 LONG
               PROGRAMABORT;                                                         01262000  02C:0264:2
```

```
       END;                                                                      01263000  02C:0265:3
  IF MPHASE = 1  AND  ABS(M[1,-1]) <= MBXTCL THEN                             6  01264000  02C:0265:3
  BEGIN                                                                          01265000  02C:0268:0
   MPHASE:= 0;                                                                6  01266000  02C:0268:3
   IF MP[2] THEN                                                                 01267000  02C:0269:1
   BEGIN                                                                         01268000  02C:0269:5
    WRITE(LINE,<"THE INDICES OF VARIABLES(WEIGHTS) IN THE MASTER ",           7  01269000  02C:026A:2
          "BASIS ARE:">);                                                        01270000  02C:026C:1
                                                                                     DATA IS 001B LONG
    WRITE(LINE,<3(/,25I5)>,WEIGHT[*]);                                           01271000  02C:026F:2
    FOR K:=2 STEP 6 UNTIL MSIZE DO                                               01272000  02C:0278:2
    WRITE(LINE,<6("W[",I5,"]",F13.6,X1)>,FOR N:=K STEP 1 WHILE                   01273000  02C:027D:1
          (N <= K+5 AND N <= MSIZE) DO [WEIGHT[N],M[N,-1]]);                     01274000  02C:0282:3
    WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",                01275000  02C:028F:5
          F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =",               01276000  02C:0291:4
          F13.6>,M[0,-1],M[1,-1]);                                               01277000  02C:0291:4
                                                                                     DATA IS 001D LONG
    WRITETODISK(0);                                                              01278000  02C:029A:2
    FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO                                           01279000  02C:029B:1
    FOR J:=1 STEP 1 UNTIL XSIZE[I] DO XVALUE[I,J]:= 0;                           01280000  02C:029F:4
    FOR J:=2 STEP 1 UNTIL MSIZE DO                                               01281000  02C:02A8:1
    BEGIN                                                                        01282000  02C:02AC:5
     I4:= - (WEIGHT[J] MOD 100);                                              8  01283000  02C:02AC:5
     IF I4 > 0 AND I4 <= NUMBLOCKS THEN                                          01284000  02C:02AF:2
     FOR J4:=1 STEP 1 UNTIL XSIZE[I4] DO                                         01285000  02C:02B0:4
     XVALUE[I4,J4]:= * + M[J,-1] * BIGX[J,J4];                                   01286000  02C:02B6:2
    END;                                                                         01287000  02C:02BC:3
    WRITE(LINE,<X5,"MASTER PROBLEM HAS A FEASIBLE SOLUTION AFTER ",           8  01288000  02C:02BD:0
          I3," ITERATIONS AND THE SOLUTIONS ARE:">,MIT);                         01289000  02C:02BE:5
                                                                                     DATA IS 0017 LONG
    FOR J:=1 STEP 1 UNTIL NUMBLOCKS DO                                           01290000  02C:02C4:2
    WRITE(LINE,<"X--VALUES",15(/,10(F13.6))>,FOR K:=1 STEP 1 UNTIL               01291000  02C:02C8:5
          XSIZE[J] DO XVALUE[J,K]);                                              01292000  02C:02CC:4
   END;                                                                          01293000  02C:02D8:5
   IF MP[3] THEN                                                              7  01294000  02C:02D8:5
   BEGIN                                                                         01295000  02C:02D9:3
    WRITE(LINE,<"THE MATRIX INVERSE FOR MASTER AT END OF PHASE I:">           7  01296000  02C:02DA:0
          );                                                                     01297000  02C:02DB:5
                                                                                     DATA IS 001D LONG
    FOR K:=0 STEP 1 UNTIL MSIZE DO                                               01298000  02C:02DE:2
    WRITE(LINE,<6(/,10(F13.6))>,M[K,*]);                                         01299000  02C:02E2:5
   END;                                                                          01300000  02C:02EB:5
  END;                                                                        7  01301000  02C:02EB:5
  IF MIT = OUTITERATION THEN WRITETODISK(1);                                  6  01302000  02C:02EB:5
  END OF MASTER;                                                                 01303000  02C:02EE:0
  IF MIT > QUITERATION  OR  MIT = OUTITERATION THEN WRITETODISK(2);           5  01304000  02C:02EE:3
  IF MP[4] THEN                                                                  01305000  02C:02F1:5
  BEGIN                                                                          01306000  02C:02F2:3
   WRITE(LINE,<"THE INDICES OF VARIABLES(WEIGHTS) IN THE MASTER ",            5  01307000  02C:02F3:0
         "BASIS ARE:">);                                                         01308000  02C:02F4:5
                                                                                     DATA IS 0011 LONG
   WRITE(LINE,<3(/,25I5)>,WEIGHT[*]);                                            01309000  02C:02F7:2
   FOR K:=2 STEP 6 UNTIL MSIZE DO                                                01310000  02C:0300:2
   WRITE(LINE,<6("W[",I5,"]",F13.6,X1)>,FOR N:=K STEP 1 WHILE                    01311000  02C:0305:1
         (N <= K+5 AND N <= MSIZE) DO [WEIGHT[N],M[N,-1]]);                      01312000  02C:030A:3
   WRITE(LINE,<"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =",                 01313000  02C:0317:5
         F13.6,/"THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =",                01314000  02C:0319:4
         F13.6>,M[0,-1],M[1,-1]);                                                01315000  02C:0319:4
                                                                                     DATA IS 001D LONG
  END;                                                                           01316000  02C:0322:2
  IF MP[5] THEN                                                               5  01317000  02C:0322:2
  BEGIN                                                                          01318000  02C:0323:0
   WRITE(LINE,<"THE MATRIX INVERSE FOR MASTER AT END OF PHASE II:">           5  01319000  02C:0323:3
         );                                                                      01320000  02C:0325:2
                                                                                     DATA IS 0017 LONG
   FOR K:=0 STEP 1 UNTIL MSIZE DO                                                01321000  02C:0328:2
   WRITE(LINE,<6(/,10(F13.6))>,M[K,*]);                                          01322000  02C:032C:5
  END;                                                                           01323000  02C:0335:5
  FOR I:=1 STEP 1 UNTIL NUMBLOCKS DO                                          5  01324000  02C:0335:5
  FOR J:=1 STEP 1 UNTIL XSIZE[I] DO XVALUE[I,J]:= 0;                             01325000  02C:033A:2
  FOR J:=2 STEP 1 UNTIL MSIZE DO                                                 01326000  02C:0342:5
  BEGIN                                                                          01327000  02C:0347:3
   INTEGER I7,                                                                5  01328000  02C:0347:3
                                                                               B.0010 IS SEGMENT 0005A
           J7;                                                                   01329000  05A:0000:1
   I7:= - (WEIGHT[J] MOD 100);                                                   01330000  05A:0000:1
   IF I7 > 0 AND I7 <= NUMBLOCKS THEN                                            01331000  05A:0002:4
```

```
          FOR J7:=1 STEP 1 UNTIL XSIZE[I7] DO                                          01332000   05A:0004:0
          XVALUE[I7,J7]:= * + M[J,-1] * BIGX[J,J7];                                     01333000   05A:0009:4
         END;                                                                            01334000   05A:000F:5
                                                                                     B.0010(05A)  IS 0012 LONG
         WRITE(LINE,<X5,"MASTER PROGRAM HAS REACHED OPTIMAL AFTER ",I2,           5      01335000   02C:0348:4
              " ITERATIONS AND THE SOLUTIONS ARE:">,MIT);                                01336000   02C:034A:3
                                                                                            DATA  IS 0012 LONG
         FOR J:=1 STEP 1 UNTIL NUMBLOCKS DO                                              01337000   02C:034F:2
         WRITE(LINE,<"X--VALUES",15(/,10(F13.6))>,FOR K:=1 STEP 1 UNTIL                  01338000   02C:0353:5
              XSIZE[J] DO XVALUE[J,K]);                                                  01339000   02C:0357:4
        END OF OBJ;                                                                      01340000   02C:0363:5
                                                                                     B.0005(02C)  IS 0365 LONG
                                                                                   STACKCODE IS SEGMENT 0005C
                                                                                  STACKCODE(05C)  IS 005A LONG
       END OF DECLA;                                                              4      01341000   00F:022B:5
                                                                                     B.0002(00F)  IS 022D LONG
                                                                                   STACKCODE IS SEGMENT 0005D
                                                                                  STACKCODE(05D)  IS 004C LONG
      END OF START;                                                               3      01342000   00A:006B:3
                                                                                     B.0001(00A)  IS 007F LONG
     EOFL:END OF PROGRAM.                                                         2      01343000   003:005E:3
                                                                                     B.0000(003)  IS 0060 LONG
                                                                                   STACKCODE IS SEGMENT 0005E
                                                                                  STACKCODE(05E)  IS 002F LONG
                                                                                            DATA  IS 001C LONG
===========================================================================================================================
NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 92.  TOTAL SEGMENT SIZE = 5226 WORDS.  CORE ESTIMATE = 9250 WORDS.  STACK ESTIMATE = 207
PROGRAM SIZE = 1343 CARDS(72 OMITTED CARDS), 7675 SYNTACTIC ITEMS, 281 DISK SEGMENTS.
PROGRAM FILE NAME: CARDOBJ.
COMPILATION TIME = 68.724 SECONDS ELAPSED; 19.481 SECONDS PROCESSING; 6.113 SECONDS I/O.
===========================================================================================================================
```

APPENDIX C

TEST DANTZIG-WOLFE (LP)
==============================

NUMBER OF OBJECTIVE FUNCTIONS = 1
NUMBER OF BLOCKS = 2
NUMBER OF ROWS IN BLOCK NO. 0 = 1
NUMBER OF ROWS IN BLOCK NO. 1 = 2
NUMBER OF ROWS IN BLOCK NO. 2 = 2
NUMBER OF COLUMNS IN BLOCK NO. 0 = 0
NUMBER OF COLUMNS IN BLOCK NO. 1 = 2
NUMBER OF COLUMNS IN BLOCK NO. 2 = 2

	OBJ. FN.	BLK. NO.	CONSTR. NAME	VARIABLE NAME	VALUE	STATUS
INPUT DATA FOR OBJ. FN.:						
	1	1		X1VAR	-4.000000	
	1	1		X2VAR	-2.000000	
	1	2		Y1VAR	-1.000000	
	1	2		Y2VAR	-2.000000	
INPUT DATA FOR MATRIX:						
		0	CUPCON	X1VAR	1.000000	
		0	CUPCON	X2VAR	4.000000	
		0	CUPCON	Y1VAR	4.000000	
		0	CUPCON	Y2VAR	2.000000	
		1	B11CON	X1VAR	1.000000	
		1	B11CON	X2VAR	2.000000	
		1	B12CON	X1VAR	2.000000	
		1	B12CON	X2VAR	1.000000	
		2	B21CON	Y1VAR	1.000000	
		2	B21CON	Y2VAR	1.000000	
		2	B22CON	Y1VAR	2.000000	
		2	B22CON	Y2VAR	4.000000	
INPUT FOR RHS OF CONSTR.:						
		0	CUPCON		18.000000	0
		1	B11CON		4.000000	1
		1	B12CON		6.000000	1
		2	B21CON		4.000000	1
		2	B22CON		10.000000	1

```
        TOLERANCE FOR PIVOT ELEMENT BEFORE MATRIX REINVERSION =                    0.
        TOLERANCE FOR R.H.S. OF CONSTRAINT BEFORE RATIO TEST =                     0.
        TOLERANCE FOR ZJ-CJ ROWS =                                                 1.000000E-05
        FLAG DENCTING THE SUPPLY OF ADVANCED SOLUTION =                            0
        MAXIMUM NUMBER OF ITERATIONS REQUIRED TO COMPUTE THE MASTER PROBLEM =      25
        NUMBER OF ITERATIONS REQUIRED TO COMPUTE BEFORE MATRIX REINVERSION =       20
        OUTPUT INTERMEDIATE RESULTS ONTO DISK AFTER THIS NUMBER OF ITERATIONS =    15
        FLAGS FOR MASTER:                          1   1   1   1   1   1   1   1
        FLAGS FOR SUBSYSTEMS #  1:                 1   1   1   1   1   1   1   1
        FLAGS FOR SUBSYSTEMS #  2:                 1   1   1   1   1   1   1   1
        @@@INPUT DATA READ IN CORRECTLY, START PROCESSING!
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =       0.000000
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =     -20.000000
    THE DUAL PRICES FOR THE COUPLING ROWS ARE:
    CUP   1      -1.000000   CUP
    THE DUAL PRICES FOR THE CONVEXITY ROWS ARE!
    CNX   1      -1.000000   CNX   2      -1.000000   CNX
         @@@@SLACK, SURPLUS AND ARTIFICIAL VARIABLES HAVE BEEN ADDED TO VARIOUS CONSTRAINS FOR SUBSYSTEM #  1
    TOTAL NUMBER CF VARIABLES IN THIS SUBSYSTEM IS =  4
    THE INDICES OF VARIABLES IN THE BASIS ARE :
       0   -1     3    4
    X[   3]      4.000000   X[   4]       6.000000   X[
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =       0.000000
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =       0.000000
        @@@SUBSYSTEM # 1 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
        AND THE LEAVING VECTOR IS X[   3]
           THE ENTERING VECTOR IS X[   2]
        THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =                8.000000
        THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =                0.000000
        THE PRODUCT OF PIVOT ELEMENTS =                               2.000000
        AND THE SUBSYSTEM IS IN PHASE = 1
    THE INDICES OF VARIABLES IN THE BASIS ARE:
       0   -1     2    4
    X[   2]      2.000000   X[   4]       6.000000   X[
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =       8.000000
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =       0.000000
    THE BASIS INVERSE AT END OF PHASE I IS AS FOLLOW:

        8.000000       1.000000       0.000000       2.000000       0.000000

        0.000000       0.000000       1.000000       0.000000       0.000000

        2.000000       0.000000       0.000000       0.500000       0.000000

        4.000000       0.000000       0.000000      -0.500000       1.000000
        @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 1 AFTER ITERATION # 2 AND THE SOLUTIONS ARE!
    THE INDICES OF VARIABLES IN THE BASIS ARE:
       0   -1     2    4
    X[   1]     X1VAR          0.000000X[   2]      X2VAR           2.000000X[
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =       8.000000
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =       0.000000
    THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW!

        8.000000       1.000000       0.000000       2.000000       0.000000

        0.000000       0.000000       1.000000       0.000000       0.000000

        2.000000       0.000000       0.000000       0.500000       0.000000

        4.000000       0.000000       0.000000      -0.500000       1.000000
        @@@@SLACK, SURPLUS AND ARTIFICIAL VARIABLES HAVE BEEN ADDED TO VARIOUS CONSTRAINS FOR SUBSYSTEM #  2
    TOTAL NUMBER CF VARIABLES IN THIS SUBSYSTEM IS =  4
    THE INDICES OF VARIABLES IN THE BASIS ARE :
       0   -1     3    4
    X[   3]      4.000000   X[   4]      10.000000   X[
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =       0.000000
    THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =       0.000000
        @@@SUBSYSTEM # 2 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
        AND THE LEAVING VECTOR IS X[   3]
           THE ENTERING VECTOR IS X[   1]
        THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =               16.000000
        THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =                0.000000
        THE PRODUCT OF PIVOT ELEMENTS =                               1.000000
        AND THE SUBSYSTEM IS IN PHASE = 1
    THE INDICES OF VARIABLES IN THE BASIS ARE:
       0   -1     1    4
    X[   1]      4.000000   X[   4]      10.000000   X[
```

```
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     16.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE I IS AS FOLLOW:

     16.000000      1.000000      0.000000      4.000000      0.000000

      0.000000      0.000000      1.000000      0.000000      0.000000

      4.000000      0.000000      0.000000      1.000000      0.000000

      2.000000      0.000000      0.000000     -2.000000      1.000000
      @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 2 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   1   4
X[  1]    Y1VAR          4.000000X[  2]    Y2VAR           0.000000X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     16.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

     16.000000      1.000000      0.000000      4.000000      0.000000

      0.000000      0.000000      1.000000      0.000000      0.000000

      4.000000      0.000000      0.000000      1.000000      0.000000

      2.000000      0.000000      0.000000     -2.000000      1.000000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  2
      @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #  1
      AND THE LEAVING WEIGHT IS W[    0]
          THE ENTERING WEIGHT IS W[ -102]
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =      4.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =     -3.000000
THE PIVOT ELEMENT =                  1.000000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  1
      @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #  1
      AND THE LEAVING WEIGHT IS W[    0]
          THE ENTERING WEIGHT IS W[ -101]
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     -5.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =     -0.750000
THE PIVOT ELEMENT =                  8.000000
THE DUAL PRICES FOR THE COUPLING ROWS ARE:
CUP   1        0.125000  CUP
THE DUAL PRICES FOR THE CONVEXITY ROWS ARE:
CNX   1       -1.000000  CNX   2       -2.000000  CNX
      @@@SUBSYSTEM # 1 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
      AND THE LEAVING VECTOR IS X[   2]
          THE ENTERING VECTOR IS X[   3]
      THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =             0.000000
      THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =             0.000000
      THE PRODUCT OF PIVOT ELEMENTS =                            0.500000
      AND THE SUBSYSTEM IS IN PHASE = 0
      @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 1 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   3   4
X[  1]    X1VAR          0.000000X[  2]    X2VAR           0.000000X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =      0.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

      0.000000      1.000000      0.000000      0.000000      0.000000

      0.000000      0.000000      1.000000      0.000000      0.000000

      4.000000      0.000000      0.000000      1.000000      0.000000

      6.000000      0.000000      0.000000      0.000000      1.000000
      @@@SUBSYSTEM # 2 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
      AND THE LEAVING VECTOR IS X[   1]
          THE ENTERING VECTOR IS X[   3]
      THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =             0.000000
      THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =             0.000000
      THE PRODUCT OF PIVOT ELEMENTS =                            1.000000
      AND THE SUBSYSTEM IS IN PHASE = 0
      @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 2 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   3   4
X[  1]    Y1VAR          0.000000X[  2]    Y2VAR           0.000000X[
```

```
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =      0.000000
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

     0.000000     1.000000     0.000000     0.000000     0.000000

     0.000000     0.000000     1.000000     0.000000     0.000000

     4.000000     0.000000     0.000000     1.000000     0.000000

    10.000000     0.000000     0.000000     0.000000     1.000000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  2
     @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #  2
     AND THE LEAVING WEIGHT IS W[    0]
          THE ENTERING WEIGHT IS W[ -202]
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =      6.500000
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE PIVOT ELEMENT =              2.000000
THE INDICES OF VARIABLES(WEIGHTS) IN THE MASTER BASIS ARE:

 -101 -202 +102
W[ -101]       1.000000 W[ -202]      0.375000 W[ -102]      0.625000 W[
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =      6.500000
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =      0.000000
     MASTER PROBLEM HAS A FEASIBLE SOLUTION AFTER   2 ITERATIONS AND THE SOLUTIONS ARE:
X--VALUES
     0.000000     2.000000     0.000000     4.000000
X--VALUES
     2.500000     0.000000     1.500000     5.000000
THE MATRIX INVERSE FOR MASTER AT END OF PHASE I:

     6.500000     1.000000     0.000000     0.250000     2.000000     0.000000

     0.000000     0.000000     1.000000     0.000000     0.000000     0.000000

     1.000000     0.000000     0.000000     0.000000     1.000000     0.000000

     0.375000     0.000000     0.000000    -0.062500     0.500000     1.000000

     0.625000     0.000000     0.000000     0.062500    -0.500000     0.000000
THE DUAL PRICES FOR THE COUPLING ROWS ARE:
CUP  1        0.000000  CUP
THE DUAL PRICES FOR THE CONVEXITY ROWS ARE:
CNX  1        0.000000  CNX  2       0.000000  CNX
     @@@SUBSYSTEM # 1 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
     AND THE LEAVING VECTOR IS X[  4]
          THE ENTERING VECTOR IS X[  1]
     THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =            11.250000
     THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =             0.000000
     THE PRODUCT OF PIVOT ELEMENTS =                            2.000000
     AND THE SUBSYSTEM IS IN PHASE = 0
     @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 1 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   3   1
X[  1]   X1VAR          3.000000X[  2]   X2VAR          0.000000X[
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE II =    11.250000
THE VALUE OF CBJECTIVE FUNCTION FOR PHASE I  =     0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

    11.250000     1.000000     0.000000     0.000000     1.875000

     0.000000     0.000000     1.000000     0.000000     0.000000

     1.000000     0.000000     0.000000     1.000000    -0.500000

     3.000000     0.000000     0.000000     0.000000     0.500000
     @@@SUBSYSTEM # 2 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
     AND THE LEAVING VECTOR IS X[  4]
          THE ENTERING VECTOR IS X[  2]
     THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =             3.750000
     THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =             0.000000
     THE PRODUCT OF PIVOT ELEMENTS =                            4.000000
     AND THE SUBSYSTEM IS IN PHASE = 0
     @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 2 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   3   2
X[  1]   Y1VAR          0.000000X[  2]   Y2VAR          2.500000X[
```

```
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =      3.750000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

    3.750000      1.000000      0.000000      0.000000      0.375000

    0.000000      0.000000      1.000000      0.000000      0.000000

    1.500000      0.000000      0.000000      1.000000     -0.250000

    2.500000      0.000000      0.000000      0.000000      0.250000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  1
    @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #   3
    AND THE LEAVING WEIGHT IS W[ -101]
        THE ENTERING WEIGHT IS W[ -301]
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     15.750000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE PIVOT ELEMENT =                  1.000000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  2
    @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #   3
    AND THE LEAVING WEIGHT IS W[ -202]
        THE ENTERING WEIGHT IS W[ -302]
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     16.090909
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE PIVOT ELEMENT =                  0.687500
THE DUAL PRICES FOR THE COUPLING ROWS ARE:
CUP   1        0.000000   CUP
THE DUAL PRICES FOR THE CONVEXITY ROWS ARE:
CNX   1        0.000000   CNX   2        0.000000   CNX
    @@@SUBSYSTEM # 1 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
    AND THE LEAVING VECTOR IS X[   3]
        THE ENTERING VECTOR IS X[   2]
    THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =              12.484848
    THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =               0.000000
    THE PRODUCT OF PIVOT ELEMENTS =                              1.500000
    AND THE SUBSYSTEM IS IN PHASE = 0
    @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 1 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1    2   1
X[  1]     X1VAR           2.666667X[  2]    X2VAR          0.666667X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     12.484848
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

   12.484848      1.000000      0.000000      0.212121      1.939394

    0.000000      0.000000      1.000000      0.000000      0.000000

    0.666667      0.000000      0.000000      0.666667     -0.333333

    2.666667      0.000000      0.000000     -0.333333      0.666667
    @@@SUBSYSTEM # 2 HAS NOT REACHED OPTIMAL AFTER ITERATION # 1
    AND THE LEAVING VECTOR IS X[   3]
        THE ENTERING VECTOR IS X[   1]
    THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =               6.272727
    THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =               0.000000
    THE PRODUCT OF PIVOT ELEMENTS =                              0.500000
    AND THE SUBSYSTEM IS IN PHASE = 0
    @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 2 AFTER ITERATION # 2 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1    1   2
X[  1]     Y1VAR           3.000000X[  2]    Y2VAR          1.000000X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =      6.272727
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

    6.272727      1.000000      0.000000      0.545455      0.409091

    0.000000      0.000000      1.000000      0.000000      0.000000

    3.000000      0.000000      0.000000      2.000000     -0.500000

    1.000000      0.000000      0.000000     -1.000000      0.500000
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  2
    @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #   4
    AND THE LEAVING WEIGHT IS W[ -302]
        THE ENTERING WEIGHT IS W[ -402]
```

```
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     16.500000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE PIVOT ELEMENT =              0.181818
THE INCOMING CANDIDATE COLUMN IS GENERATED FROM SUBSYSTEM #  1
      @@@MASTER PROBLEM HAS NOT REACHED OPTIMAL AFTER ITERATION #   4
      AND THE LEAVING WEIGHT IS W[ -102]
              THE ENTERING WEIGHT IS W[ -401]
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     17.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE PIVOT ELEMENT =              1.166667
THE DUAL PRICES FOR THE COUPLING ROWS ARE:
CUP  1        0.000000  CUP
THE DUAL PRICES FOR THE CONVEXITY ROWS ARE:
CNX  1        0.000000  CNX  2        0.000000  CNX
      @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 1 AFTER ITERATION # 1 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   2   1.
X[  1]   X1VAR           2.666667X[  2]    X2VAR           0.666667X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     12.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

     12.000000     1.000000     0.000000    -0.000000     2.000000

      0.000000     0.000000     1.000000     0.000000     0.000000

      0.666667     0.000000     0.000000     0.666667    -0.333333

      2.666667     0.000000     0.000000    -0.333333     0.666667
      @@@OPTIMAL HAS REACHED FOR SUBSYSTEM # 2 AFTER ITERATION # 1 AND THE SOLUTIONS ARE:
THE INDICES OF VARIABLES IN THE BASIS ARE:
   0  -1   1   2
X[  1]   Y1VAR           3.000000X[  2]    Y2VAR           1.000000X[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =      5.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE BASIS INVERSE AT END OF PHASE II IS AS FOLLOW:

      5.000000     1.000000     0.000000    -0.000000     0.500000

      0.000000     0.000000     1.000000     0.000000     0.000000

      3.000000     0.000000     0.000000     2.000000    -0.500000

      1.000000     0.000000     0.000000    -1.000000     0.500000
THE INDICES OF VARIABLES(WEIGHTS) IN THE MASTER BASIS ARE:

 -301 -402 -401
W[ -301]      0.571429 W[ -402]      1.000000 W[ -401]      0.428571 W[
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE II =     17.000000
THE VALUE OF OBJECTIVE FUNCTION FOR PHASE I  =      0.000000
THE MATRIX INVERSE FOR MASTER AT END OF PHASE II:

     17.000000     1.000000     0.000000     0.000000    12.000000     5.000000

      0.000000     0.000000     1.000000     0.000000     0.000000     0.000000

      0.571429     0.000000     0.000000    -0.428571     2.285714     6.000000

      1.000000     0.000000     0.000000     0.000000     0.000000     1.000000

      0.428571     0.000000     0.000000     0.428571    -1.285714    -6.000000
      MASTER PROGRAM HAS REACHED OPTIMAL AFTER  5 ITERATIONS AND THE SOLUTIONS ARE:
 X--VALUES
      2.857143     0.285714     0.571429     0.000000
 X--VALUES
      3.000000     1.000000     0.000000     0.000000
```

APPENDIX D

APPENDIX D

LIST OF POSSIBLE ERROR MESSAGES

a) CONTROL CARD 'NUMBLK' IS EXPECTED OR THE NUMBER OF BLOCKS IN THE STRUCTURE IS NOT PROPERLY SPECIFIED.

b) CONTROL CARD 'NUMROW' IS EXPECTED.

c) CONTROL CARD 'NUMCOL' IS EXPECTED.

d) INCORRECT OBJECTIVE FUNCTION NUMBER ON DATA CARD UNDER CONTROL CARD 'OBJCOF'.

e) DUPLICATE VARIABLE NAME SPECIFIED FOR OBJECTIVE FUNCTION NO. 1, BLOCK NO. 23, UNDER CONTROL CARD 'OBJCOF'.

f) MORE DATA ARE EXPECTED FOR OBJECTIVE FUNCTION NO. 2, BLOCK NO. 14 UNDER CONTROL CARD 'OBJCOF'.

g) TOO MANY DATA SPECIFIED FOR OBJECTIVE FUNCTION NO. 3, BLOCK NO. 11 UNDER CONTROL CARD 'OBJCOF'.

h) TOO MANY BLOCKS SPECIFIED FOR OBJECTIVE FUNCTION NO. 4, UNDER CONTROL CARD 'OBJCOF'.

i) CONTROL CARD 'OBJCOF' IS EXPECTED.

j) INCORRECT BLOCK NUMBER ON DATA CARD UNDER CONTROL CARD 'MATRIX'.

k) TOO MANY DATA OR WRONG VARIABLE NAME SPECIFIED FOR CONSTRAINT CUT256, IN BLOCK NO. 18, UNDER CONTROL CARD 'MATRIX'.

l) MORE CONSTRAINTS ARE EXPECTED FOR BLOCK NO. 17, UNDER CONTROL CARD 'MATRIX'.

m) TOO MANY DATA OR WRONG VARIABLE NAME SPECIFIED FOR CONSTRAINT TRK144, IN BLOCK NO. 12, UNDER CONTROL CARD 'MATRIX'.

n) CONTROL CARD 'MATRIX' IS EXPECTED.

o) INCORRECT BLOCK NUMBER ON DATA CARD UNDER CONTROL CARD 'RHS '.

p) DUPLICATE CONSTRAINT NAME USED IN BLOCK NO. 15, UNDER CONTROL CARD 'RHS '.

q) MORE CONSTRAINTS ARE EXPECTED FOR BLOCK NO. 19, UNDER CONTROL CARD 'RHS '.

r) INVALID STATUS SPECIFIED FOR CONSTRAINT CREW32, IN BLOCK NO. 7, UNDER CONTROL CARD 'RHS '.

s) WRONG CONSTRAINT NAME USED FOR BLOCK NO. 5, UNDER CONTROL CARD 'RHS '.

t) TOO MANY CONSTRAINTS SPECIFIED FOR BLOCK NO. 16, UNDER CONTROL CARD 'RHS '.

u) CONTROL CARD 'RHS ' IS EXPECTED.

APPENDIX E

APPENDIX E

COMPARISON OF THE SIMPLEX AND REVISED SIMPLEX METHOD

	SIMPLEX	REVISED SIMPLEX
1.	Less arithmetical operations.	More arithmetical operations.
2.	More storage space is required for the whole simplex tableau.	Less storage is used as only the matrix for basis inverse is stored.
3.	No preservation of original matrix.	Preservation of original matrix thus reinverting of matrix for basic inverse is possible.
4.	Use of dense matrix.	Use of sparse matrix.
5.	Unstable with respect to accumulation of rounding-off.	Better control in rounding-off error accumulation.